aerospace

MDPI

Article

Formulating a Learning Assurance-Based Framework for AI-Based Systems in Aviation

Friedrich Werner [1], Johann Maximilian Christensen [2,†,*], Thomas Stefani [1,†], Frank Köster [2], Elena Hoemann [2] and Sven Hallerbach [2]

[1] Institute for AI Safety and Security, German Aerospace Center (DLR), 89081 Ulm, Germany; thomas.stefani@dlr.de (T.S.)
[2] Institute for AI Safety and Security, German Aerospace Center (DLR), 53757 Sankt Augustin, Germany
* Correspondence: johann.christensen@dlr.de
† These authors contributed equally to this work.

Abstract

The European Union Aviation Safety Agency (EASA) is developing guidelines to certify AI-based systems in aviation with learning assurance as a key framework. Central to the learning assurance are the definitions of a Concept of Operations, an Operational Domain, and an AI/ML constituent Operational Design Domain (ODD). However, because no further guidance on these concepts is provided to developers, this work introduces a framework for defining them. For the concepts of the Operational Domain of the overall system and the AI/ML constituent ODD, a tabular definition language is introduced. Furthermore, processes are introduced to define the different necessary artifacts. During the specification process for the AI/ML constituent ODD, existing steps were identified and consolidated, including the identification of domain-specific concepts for the AI/ML constituent. To validate the framework, it was applied to the PYCASX system, which employs neural-network-based compression. For this use case, the framework produced an AI/ML constituent ODD with finer detail than other ODDs defined for the same airborne collision avoidance use case. Thus, the proposed novel framework is an important step toward a holistic approach aligned with EASA's guidelines.

Keywords: AI Engineering; W-Shaped Process; ConOps; OD; ODD; Model-Based Systems Engineering; Aviation; AI Certification; Safety-by-Design

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) have become increasingly capable of solving a wide range of tasks and are therefore being adopted by an increasing number of industries in recent years. One of those industries is aviation, an industry governed by one of the strictest certification processes. Still, there are no certification processes available for AI-based systems in aviation, as it has been identified that existing standards, such as ARP4754A, DO-178, or DO-200B, are insufficient for the development of AI-based systems [1–4]. These shortcomings in existing standards stem from the paradigm shift from requirements-driven to data-driven development for AI-based systems [5–7]. To address this issue and establish a certification process for these systems, EASA, in cooperation with other stakeholders, is developing a guidance framework to ensure the trustworthiness of AI-based systems [5,8]. EASA identified four building blocks necessary to achieve their AI trustworthiness concept and to enable the use of AI-based systems in aviation, namely,

Check for updates

Academic Editor: Graham Wild

Received: 10 November 2025
Revised: 30 January 2026
Accepted: 18 February 2026
Published: 19 February 2026

the AI trustworthiness analysis, the AI assurance, the Human factors for AI, and the AI safety risk mitigation [8].

The most important component of an AI-based system that distinguishes it from traditional software systems is the AI/ML constituent, which includes the ML inference model. For this AI/ML constituent, an Operational Design Domain (ODD) must be defined to capture the operating conditions under which the AI/ML constituent must operate as expected [5]. Furthermore, this AI/ML constituent ODD must also provide "a framework for the selection, collection, and preparation of the data" [5], which is used to develop the ML inference model. The AI/ML constituent ODD is therefore crucial, as it is a key component of the developer's argumentation in the learning assurance process, which is part of AI assurance. EASA developed concepts, such as the AI/ML constituent ODD, to enable the Safety-by-Design development of AI-based systems. For these concepts, EASA outlines the objectives the developer must meet when developing an AI-based system. However, there are currently no regulatory guidelines on the tools and methods to be used to achieve the various objectives associated with these concepts.

While the framework proposed by EASA sets high-level objectives for AI-based system development, it provides only limited guidance on how system-level operational assumptions can be systematically translated into learning assurance artifacts for an AI/ML constituent. In particular, a key remaining challenge in developing safe AI-based systems using EASA's framework, especially the learning assurance process, is how to define the Operational Domain (OD), the AI/ML constituent ODD, and subsequently, the neural networks that are part of the AI/ML constituent, and how to ensure traceability between these artifacts. To address this gap, this work aims to research and apply Safety-by-Design methodologies to bridge the gap between AI trustworthiness analysis and AI assurance, and to help developers of AI-based systems achieve the objectives of learning assurance, focusing on the AI/ML constituent ODD [5]. Furthermore, how to achieve the different objectives outlined in the AI trustworthiness analysis and AI assurance remains an open question, as EASA does not specify the tools and processes to be used [5]. Therefore, this work also contributes a set of processes and formalisms that enable developers to achieve the objectives of EASA's framework [5], as listed in Table 1.

Based on these objectives, a multi-step, artifact-driven framework is proposed to bridge the gap between AI trustworthiness analysis and AI assurance. Therefore, the main focus of this work is to define a Concept of Operations, an Operational Domain, and a functional decomposition for the AI-based system, and to derive the AI/ML constituent Operational Design Domain from these artifacts. Additionally, the implications of these artifacts for the AI/ML constituent architecture are investigated. The usability of the introduced frameworks is demonstrated through the collision-avoidance use case, which illustrates how the frameworks support developers in developing AI-based systems aligned with future EASA regulatory requirements. By answering these open questions, a set of processes and tools is introduced that developers can use to build an AI-based system compliant with EASA regulatory guidelines.

This paper describes various processes for specifying an OD and an AI/ML constituent ODD, including necessary formalisms. Section 2 introduces the state-of-the-art on the introduced research question. In Section 3, the proposed framework is presented, and in Section 4, the case study for the framework validation is introduced. Section 5 discusses the findings, and Section 6 concludes.

Table 1. The objectives of EASA's trustworthiness frameworks [5] that are considered for the framework of this work.

Objective	Description
CO-01	"The applicant should identify the list of end users that are intended to interact with the AI-based system, together with their roles, their responsibilities [. . .] and expected expertise [. . .]."
CO-02	"For each end user, the applicant should identify which goals and associated high-level tasks are intended to be performed in interaction with the AI-based system."
CO-04	"The applicant should define and document the ConOps for the AI-based system, including the task allocation pattern between the end user(s) and the AI-based system. A focus should be put on the definition of the OD and on the capture of specific operational limitations and assumptions."
CO-06	"The applicant should perform a functional analysis of the system, as well as a functional decomposition and allocation down to the lowest level."
DA-03	"The applicant should define the set of parameters pertaining to the AI/ML constituent ODD, and trace them to the corresponding parameters pertaining to the OD when applicable."
DA-06	"The applicant should describe a preliminary AI/ML constituent architecture [...]"
LM-01	"The applicant should describe the ML model architecture."

2. State of the Art

In aviation, safety-critical systems and components must be certified before deployment. In Europe, this is the responsibility of the European Union Aviation Safety Agency [9]. EASA specifies the regulations that have to be followed and publishes advisories on acceptable means of compliance and guidance material. Under those regulations, developers can provide evidence that the system or component complies with the applicable requirements. The development of acceptable means of compliance and guidance material is supported by the development of industry standards by agencies such as the European Organization for Civil Aviation Equipment (EUROCAE) [10]. As shown in Figure 1, a key aviation standard is ARP4754A/ED-79A, which defines the system development process using the V-model. One important input to the V-model is the safety assessment, which must be conducted in accordance with ARP4761A [11]. For traditional software development, the relevant guidance standard is DO-178/ED-12 [2,12]; for hardware development, DO-254/ED-80 [13,14] is an accepted means of compliance.

However, with the shift to data-driven development for AI-based systems, existing standards, along with their corresponding regulatory guidance and rulemaking, are insufficient for development and certification [4]. To address this issue, various regulatory agencies launched initiatives and research projects to close identified gaps [8,15–18]. An overview of recent advances across topics such as neural network verification can be found in the literature [19,20]. One important step in that direction is the EASA Concept Paper: Guidance for Level 1 & 2 machine learning applications [5]. Here, EASA outlines the technical objectives and organizational provisions it considers necessary to approve AI-based systems for what it defines as level 1 and 2 applications. Furthermore, this concept paper is receiving iterative updates and will be extended to level 3 AI-based systems. EASA defines AI levels based on the increasing autonomy of AI-based systems. While level 1 offers assistance to humans, level 2 applies to cooperation and collaboration between humans and AI-based systems, up to shared responsibilities. Finally, AI-based systems defined as level 3 are autonomous in their decision-making and implementation [8]. Two key aspects introduced in the concept paper are AI trustworthiness analysis and AI assurance. The AI trustworthiness analysis contains a characterization and different assessments of the AI-based system. This includes a Concept of Operations (ConOps) that provides a refined definition of the AI-based system and its operating environment. The description of the operating environment is formalized as the Operational Domain (OD). The notion to specify the operation conditions for an AI-based system has been developed by the automotive domain [21–27] and has already been adapted by the maritime [28] and railway [29] domains. In the automotive domain, the concept used to describe system-level operating conditions is called ODD. In contrast, the EASA naming convention uses the term Operational Domain (OD) for a similar purpose. This difference in naming convention is important to acknowledge, as EASA uses the term ODD at the AI/ML constituent level, where it denotes a different concept. The concept of the AI/ML constituent ODD is described in greater detail in the next paragraph. In the automotive domain, a key property of the specified operating environment is its hierarchical, taxonomy-based structure [21,22,30–32]. This structure enables standardization [21,22] of the conditions to be identified, as well as strategies [24,27,33] for identifying them in the operating environment. How to specify such an OD for an AI-based system in aviation has been previously explored in different domains such as air traffic management [34,35], airborne collision avoidance [36–41], and unmanned aircraft [42].

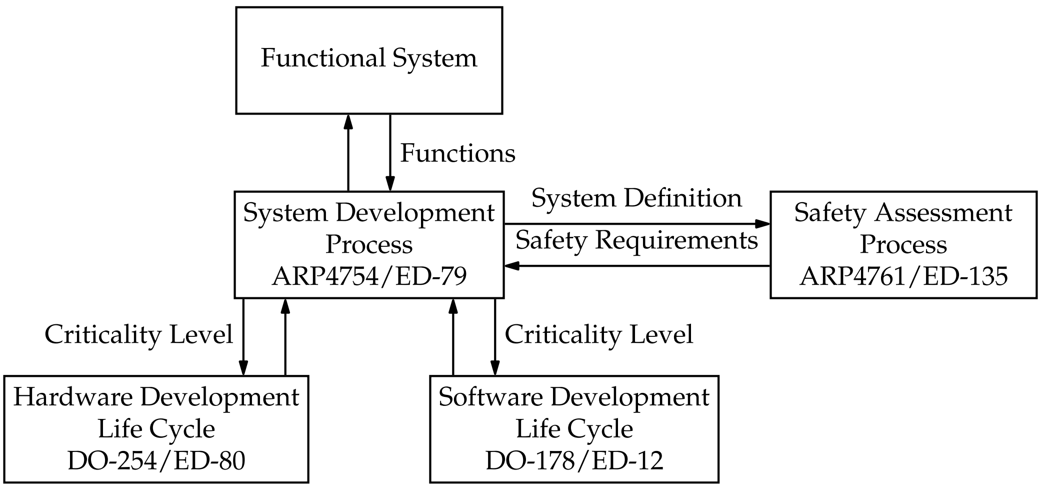

Figure 1. Important standards used in the development of an aviation system to provide evidence to meet the regulatory requirements. Adapted from [19].

With AI assurance, EASA provides specific guidance required for developing an AI-based system [5]. This includes the development and post-ops AI explainability, as well as learning assurance. Learning assurance is a new concept introduced by EASA to define a development assurance method for AI/ML constituents. EASA defines learning assurance as all actions required to ensure, with a sufficient level of confidence, that errors are identified and corrected in the learning process to meet the required level of performance for the AI/ML constituent. To implement learning assurance, EASA proposes its W-shaped process [5]. While other works from different domains [7,10,43–45] have already introduced processes or frameworks with similar ideas, the W-shaped process introduced here adapts and extends them to fit within EASA's framework. Furthermore, additional novel concepts were introduced in the W-shaped process, as reported in other works [5,46]. One such important novel concept is the AI/ML constituent ODD. With the AI/ML constituent ODD, a developer must define the operating conditions under which the AI/ML constituent is expected to operate as intended [5]. This AI/ML constituent ODD must include finer detail than the system-level operational domain to enable its use in data and learning management. Specifying an AI/ML constituent ODD is important, as the defined attributes and ranges inform data-quality requirements and verify that the dataset used to develop the ML inference model is complete and representative of the expected operating conditions of the AI/ML constituent in later use. Furthermore, this AI/ML constituent ODD is used to estimate the ML inference model's generalization capabilities and to define the AI/ML constituent's behavior when exposed to out-of-distribution data. This AI/ML constituent ODD concept is now data-centric [6]; therefore, its specification must include data-specific considerations [29,47,48]. These data-specific considerations can include how system-level parameters are mapped to the AI/ML constituent ODD [47] or how to represent the data distributions for the AI/ML constituent ODD's parameters [48]. In addition, different methodologies were introduced [29,47,48] to enable the identification and definition of parameters for an AI/ML constituent ODD that includes these data-specific considerations. Importantly, EUROCAE [49] and the G34 working group of SAE [50] are already developing standards that use a similar notation for the AI/ML constituent ODD, as introduced by EASA. Most notably, ED-324 and ARP6983 [51], which will introduce a process for the certification and approval of AI in safety-related products. Both standards are expected to be first published in 2026 [52].

A current focus of AI-based system development in aviation is the use case of airborne collision avoidance [37,39,40,53,54]. One system involved in this use case is ACAS X. This system is designed to generate resolution advisories for pilots to follow to avert a potential mid-air collision between two aircraft. However, ACAS X relies on precalculated data to generate resolution advisories, which, given its size, is challenging to store on current and certified avionics hardware. Therefore, one approach is to use neural networks to reduce the memory footprint of ACAS X [53,54]. However, relying solely on neural networks does not reduce the memory footprint in a loss-free manner [53,54]. Therefore, the Safety Net concept, as a hybrid architecture for data compression, was proposed [36,54,55]. This concept's advantage is that it can reduce the memory footprint while ensuring the correctness of the compressed data [36,54,55].

3. Proposed Framework

In developing the novel framework, the considered objectives are organized into five sequential, partially iterative steps. These steps are ordered first by the objectives each step fulfills and, second, by the framework within which these objectives are situated. The introduced steps aim to meet the objectives of two EASA trustworthiness guidelines frameworks: AI trustworthiness analysis and AI assurance. Notably, each step produces a well-defined output that serves as input to the subsequent steps. The first three steps are the ConOps and OD definitions, as well as a functional decomposition of the system. These three steps yield artifacts: operational scenarios, a tabular OD specification, and a functional allocation that identifies AI/ML-based constituents. The next step is to define the AI/ML constituent ODD based on the system-level artifacts produced during the AI trustworthiness analysis. This AI/ML constituent ODD must contain the ranges and distributions of the operating parameters for which the AI/ML constituent is designed to operate. The final step provides guidance on how the specified AI/ML constituent ODD can influence the AI/ML constituent architecture, with a focus on input feature selection. While this step does not directly achieve a specific objective, it provides a necessary foundation for other objectives in data preparation within data management [5]. Importantly, these two steps are based on the artifacts produced by the AI trustworthiness analysis. Furthermore, these are part of the learning assurance within the AI assurance framework, which is used to develop the AI/ML constituent [5]. Following these steps should help the developers build an assurance case that the AI-based system complies with a defined level of performance and the defined requirements [5]. In summary, the main objective is the introduction of a framework that bridges the gap between the AI trustworthiness analysis and the AI/ML constituent ODD that is part of the AI assurance. Importantly, using the proposed framework, only a subset of all objectives of EASA's guidelines is fulfilled. Therefore, the proposed framework will not be sufficient to build a complete assurance case for an AI-based system. Nevertheless, fulfilling the considered objectives is a mandatory prerequisite when building an assurance case according to EASA [5].

3.1. Definition of a Concept of Operations

The first step in developing any AI-based system is to define it through a ConOps to identify all users and specify the system's capabilities and limitations [5]. Based on EASA's suggested objectives CO-01 and CO-02 for the system, all end users who interact with the AI-based system [5] must be identified and documented. Furthermore, this documentation also includes the goals and high-level tasks a user intends to perform when interacting with the AI-based system [5]. The ConOps documents the characteristics of the AI-based system from the users' operational viewpoint [5]. Therefore, this methodology proposes the five steps for the definition of the ConOps based on the ISO/IEC/IEEE 29148:2018 [56]. These steps consist of a description of the current system, the justification for and nature of the changes, the description of the proposed AI-based system, the definition of the task allocation pattern, and lastly, the description of operational scenarios. The definition of the operational environment is replaced by the operational domain, which is defined in the following subsection. The artifact produced by this step is a ConOps description for the AI-based system.

As the first step, a description of the current system must be created [56]. This description must include an overview of the system's functionality, an explanation of the underlying technology, and a list of aviation standards applicable to the system. This allows other stakeholders to better understand the current state of the problem domain [56] and also defines the initial scope for the system's intended use. As recommended in ISO/IEC/IEEE 29148:2018 [56], the justification of changes must highlight the shortcomings of the current system or situation. Furthermore, the nature of the proposed changes has to be stated [56]. Importantly, the change introduced by the AI component must be stated to communicate the expected effect on AI component usage.

Based on these steps, a description of the AI-based system must be created. The description of the proposed AI-based system must contain "[t]he operational environment and its characteristics" [56]. Furthermore, it must also include the capabilities and functions provided by the proposed system [56]. This must also include the major system components needed for those capabilities and functions [56]. Lastly, the description also provides the task allocation and interactions between the end users and the AI-based system [5]. The description must enable all stakeholders to clearly understand the AI-based system's functionality and responsibilities. Moreover, the description must be supplemented with operational scenarios. Each scenario description must comprise multiple steps that specify the environmental conditions, the individual system functions, and the task allocation between the end user and the AI-based system required to achieve the higher-level task or goal in the scenario [5]. In addition, these steps must be described so that each stakeholder, regardless of technical background, can understand the system's functionalities, responsibilities, and limitations. This is important because understanding the AI-based system, especially from the end user's perspective, will be the foundation of trust in the system [5]. Finally, these scenario descriptions must also include scenarios where the AI-based system is outside its designed operating conditions [5] to highlight the system's fallback measures and limitations.

3.2. Definition of an Operational Domain

The OD of an AI-based system describes the operating conditions under which it is designed to function as expected [5]. Furthermore, the OD must be in accordance with the defined ConOps for the AI-based system [5]. While EASA formalized properties of the OD concept, it did not provide guidelines for specifying an OD in a formal manner. The automotive sector has already advanced the field of capturing operational conditions by standardizing various aspects [21,22,57] and has successfully applied these standards to the development of a level 3 automated driving system [58]. Therefore, this approach leverages concepts from other domains to develop the OD concept for aviation. To specify the OD for an AI-based system, Figure 2 introduces different concepts and their relations, building on the concepts and relationships of an OD in the automotive domain [59] but adapted such that the terminology fits the concepts and definitions of EASA. The taxonomy concept characterizes operational conditions by classifying them using a set of attributes [22]. These attributes are used in the statements to define the individual ranges of the operational conditions, as shown at the bottom right of Figure 2. A statement describes an operational condition that is either excluded or included in the OD specification. The collection of statements constitutes the OD specification for the AI-based system, i.e., the operating conditions under which it is expected to operate. Importantly, the OD is typically a subset of all possible operational conditions, as systems may be constrained by their function and design [59]. For the statements that define the OD, a specification language is needed, as it enables communication and understanding of the OD by different stakeholders [22]. Therefore, to specify an OD for an AI-based system, both an OD taxonomy and an OD specification language have to be defined [59]. The definition of an OD taxonomy is specific to the use case for which the AI-based system is built, while the OD specification language should be usable for any use case. These two components define a concrete OD for an AI-based system. The following sections will first introduce a hierarchical structure for the OD taxonomy and a definition language for the OD. Secondly, an approach to specifying a concrete OD for an AI-based system's use case is outlined.

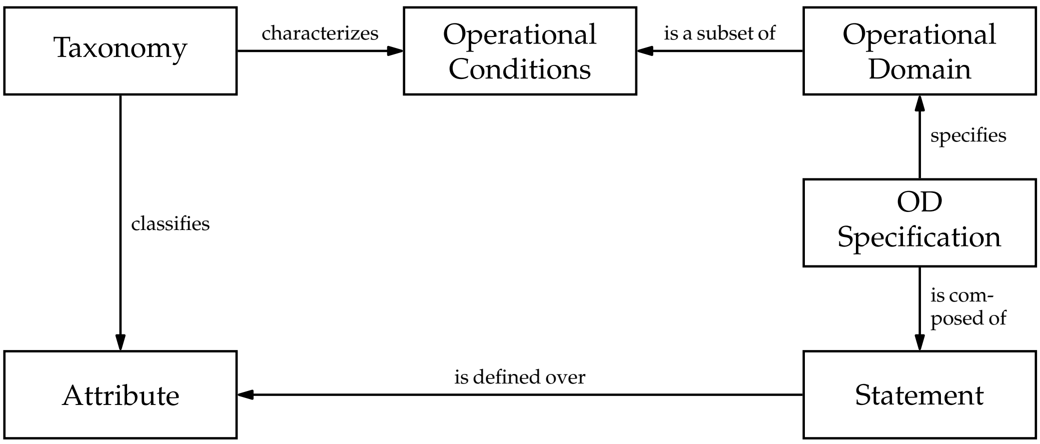

Figure 2. Relation between the different concepts connected to an operational domain. The figure is adapted from [59], which defined the relations and components for the automotive concept of an OD.

3.2.1. Operational Domain Taxonomy

An OD taxonomy defines the attributes that can comprise the operating environment of an AI-based system and organizes them in a hierarchical structure [22]. Thus, the taxonomy must cover all attributes necessary to define the elements and conditions of the operating environment relevant to the AI-based system. This is important because an OD is considered complete only if no safety-relevant attributes are missing for the system's defined ConOps [60]. Completeness is therefore an important prerequisite for arguing a system's safety based on the OD [60]. However, given the diversity of AI-based system use cases in the aviation sector [8,16,17], developing a single, universal taxonomy that accommodates all use cases remains a significant challenge. While individual domains may share similar taxonomies for different use cases, a single universal OD taxonomy applicable across domains is not feasible. Consequently, the proposed framework does not aim to define a universally complete taxonomy; instead, it provides a generic, risk-driven, and system-oriented approach to establish a foundation for OD taxonomy definition applicable to various use cases, as further described in Section 3.2.3. For the basic structure of the OD taxonomy, three top-level attributes are proposed, namely *scenery elements*, *environmental elements*, and *dynamic elements* [22,61]. This proposed top-level attribute structure has already been successfully applied to air traffic management use cases [34] and air operations use cases [39]. The first top-level attribute, scenery elements, should include all attributes that define elements that can be considered spatially fixed within the operating environment of the AI-based system [22]. Next, the environmental elements should include all attributes that describe weather and atmospheric conditions, as well as other attributes considered non-scenery elements [22]. A non-scenery element can, for example, be the system's connectivity to external infrastructure [22]. Finally, the dynamic elements encompass all attributes of other participants in the environment and of the AI-based system itself [22]. The other participants can be seen as agents that can move or change in the operating environment of the AI-based system. Additionally, the dynamic elements should include attributes that may restrict the performance or capabilities of the AI-based system itself [22]. This outlined taxonomy classification provides only a high-level guideline for the attributes a developer must define for their use case, enabling its application across a wide variety of aviation use cases. For the definition of an OD taxonomy, a variety of formats can be used, including, among others, textual descriptions [21,22] or block definition diagrams written in SysML [34,39].

3.2.2. Operational Domain Definition Language

The OD definition language is an important component of an OD definition, as it enables developers to capture operational conditions consistently, accurately, and clearly for different stakeholders. The latter is required, as EASA views the OD as part of the ConOps, which must be comprehensible from a user's perspective [5]. In the automotive domain, the OD definition language is already standardized by the ISO 34503:2023 [22]. Therefore, for the definition language, a syntax based on the tabular approach of the ISO 34503:2023 is introduced. This format was introduced because it enables the OD to be accessible to all stakeholders connected to the system. The tabular format comprises columns for the top-level attribute, multiple levels of sub-attributes, the qualifier, and the attribute with its corresponding value and unit. The columns of the top-level attribute, the sub-attributes, and the attribute are based on the hierarchical taxonomy. The qualifier column indicates whether an attribute and its values are included or excluded from the OD. If an attribute and its value are included, the system can function under the stated condition. This also includes all possible combinations of the other included attributes and their attribute values. If the qualifier is excluded, the attribute and its value constitute a condition under which the system is specified not to function. Therefore, in such situations, the system is not allowed to be operated or must be deactivated. Attributes not listed in the OD specification are assumed not to affect the AI-based system [22]. These assumptions must subsequently be validated to ensure their validity. The value of the attribute is described in the column

"Attribute value". Depending on the attribute type, a wide range of styles may be required to represent the value. The advantage of this approach is that the tabular format is not only human-readable but also easily machine-readable.

3.2.3. OD Specification

For the specification of an OD, the *domain-agnostic and risk-based OD definition approach* [61] is used but adapted to fit the framework proposed by EASA [5]. The approach was chosen as it defines the OD definition process domain independently. This is an advantage over other processes [24,62], which primarily focus on the automotive domain. As EASA specified certain aspects regarding the OD, these are incorporated into the existing process. Furthermore, the existing process is extended to include additional guidance on identifying and selecting attributes and their value ranges. The introduced process consists of three steps: initialization, refinement, and validation and verification. The initialization starts with defining the taxonomy for the use case, following the structure outlined in Section 3.2.1. This taxonomy initialization is based on the ConOps for the AI-based system, which provides a generic description of the system and concrete descriptions of scenarios in which it will be operated. Both descriptions can be used to identify the taxonomy attributes. To identify the different elements under the operational conditions of individual scenarios, approaches similar to the 6-layer model [33], developed for the automotive domain, can be used. The 6-layer model decomposes the operating environment into multiple predefined layers and uses this decomposition to systematically identify all relevant elements [33]. What each layer describes must be adapted to the individual use cases and the applied model. In addition, depending on the use case, standardization efforts may already provide an initial set of attributes for an OD taxonomy. Once an initial taxonomy is defined, it can be used to specify an OD for the system, followed by assigning values to all relevant attributes. For this initial assignment of values, the ConOps for the AI-based system should be used. An important step in determining an attribute range is identifying the qualifier. This allows determining whether a taxonomy attribute is relevant to the AI-based system's operation and whether it is included or excluded from the system's operating conditions.

If an initial OD is defined, it builds the basis for the refinement. This refinement comprises multiple tasks, similar to prior work [34,61]. First, based on the individual attributes identified and established in the ConOps scenarios, determine whether these attributes require clarification in the scenarios. This is required due to the close connection between the ConOps and OD scenarios and the requirement that they be consistent [5]. Secondly, standards associated with the use case are analyzed. Depending on the proposed AI-based system, standards may already have defined operational services and environment descriptions, or minimum design and performance requirements. Thirdly, subject-matter experts can refine the defined OD taxonomy or specification.

Importantly, as the final step of this refinement, the consistency between the OD and ConOps must again be checked, as newly identified attributes in the refinement may introduce additional operational conditions for the system. These newly identified operational conditions may need to be reflected in the ConOps operational scenario descriptions. This can lead to adapting the already defined scenarios or introducing new scenario descriptions that include the newly identified operational conditions. The need for adaptation is a bidirectional relationship between the ConOps and OD specifications. Therefore, if the ConOps is adapted, it requires corresponding adaptations to the OD taxonomy or specification. Adaptations such as these are necessary; otherwise, inconsistencies would persist between the OD and ConOps for the AI-based system.

If the refinement is complete, a verification and validation of the specified OD is necessary. This verification and validation should ensure that the OD describes all necessary operating conditions to meet the system requirements and that the OD and ConOps are consistent. After verification and validation of the specified OD against the defined ConOps and operational assumptions, the OD serves as a formal input to the downstream functional decomposition and subsequently to the AI/ML constituent ODD specification.

3.3. Functional Decomposition of the AI-Based System

To achieve a functional decomposition of the AI-based system, a two-step approach is proposed. The first step is a functional analysis of the system, and the second step is the definition of the preliminary system architecture. At the end of the functional decomposition, functions should be allocated to the AI/ML constituents. The goal of functional analysis is to identify the functions implemented by the AI-based system. As described by the ARP4754 [1], a function should capture the behavior of a system, regardless of the chosen implementation. The functions of a system can be captured by a functional tree [63]. The analysis starts with the system's high-level function, which is then decomposed into lower-level functions [63]. These lower-level functions are further decomposed until an atomic function is reached, one that cannot be split further, forming the foundation of the system's functional requirements. To create individual functions, a set of rules is recommended [63]. Most importantly, these atomic functions should be described as generally as possible, using a verb and a noun, to avoid restricting the variety of solutions [63]. In the second step, a preliminary system architecture is introduced. This includes an overview of all system components and shows the information flow among them [5]. One requirement of EASA for creating an AI-based system architecture is labeling the components, functions, and items, whether AI/ML-based or not [5]. A function is AI/ML-based if it is implemented by an item that contains an AI/ML constituent. This classification is important to specify, as it highlights which parts of the system the AI assurance applies to. In addition, processes such as ARP4754 [1] or ARP4761 [64] can identify additional functional, safety, and security requirements. Based on the functional decomposition from the previous step and the additional requirements identified, a foundation for the AI/ML constituent requirements is established. However, as noted in AIR6988 [4], the ARP4754 [1] guideline has gaps when assessing AI/ML-based systems. Therefore, when dealing with such systems, ARP4754 [1] may be insufficient, and additional processes, such as ARP6983 [51], must be applied. The functional allocation explicitly identifies AI/ML-based constituents, thereby defining the scope of subsequent learning assurance activities.

3.4. Definition of the AI/ML Constituent Operational Design Domain

At the AI assurance level, EASA [5] requires further refinement and the allocation of system-level requirements to the AI/ML constituents. An important concept at the AI/ML constituent level is the definition of the AI/ML constituent ODD. Furthermore, the AI/ML constituent ODD must define constraints and requirements for the ML inference model and the data used to build and deploy [5]. Lastly, the AI/ML constituent ODD must also incorporate constraints and requirements on the data that the ML inference model will be exposed to during inference operations [5]. As the OD and the ODD are conceptually similar, the proposed framework is built upon the OD framework introduced in Section 3.2. Importantly, due to key differences between the OD and ODD, several adaptations must be incorporated into the framework for the ODD definition to comply with EASA requirements [5]. Furthermore, the AI/ML constituent ODD refines the attributes, parameter ranges, and operational assumptions specified in the ConOps, the system-level OD, and the functional decomposition produced in the previous steps. These upstream artifacts are explicitly included to ensure traceability between AI trustworthiness and AI assurance artifacts.

3.4.1. AI/ML Constituent ODD Taxonomy

Similar to the OD taxonomy at the system level, the AI/ML constituent ODD taxonomy defines the attributes of the operating environment for the AI/ML constituent [5]. Therefore, a similar approach was used to define the attributes for the AI/ML constituent ODD, as was used for the OD. This includes the structuring of the operating conditions into the top-level attributes of *scenery elements*, *environmental elements*, and *dynamic elements*. As the operating conditions for the AI/ML constituent are specified at the subsystem level [5], influences stemming from the overall system can affect the provided data. As such, these influences must be captured in the AI/ML constituent ODD to ensure that the data conforms to the specified conditions [5,65]. Accordingly, the top-level attributes are extended with the top-level group of *operating parameters* [5]. This set of attributes must include all additional operating parameters introduced by other system components with which the AI/ML constituent interacts.

3.4.2. AI/ML Constituent ODD Definition Language

The OD definition language is also applied to the AI/ML constituent ODD. Therefore, the same tabular structure as in Section 3.2.2 is used, but extended by two additional columns. First, for each attribute, the data distribution must be defined [5]. Whenever possible, the system component (e.g., a sensor) that generates the data should also be recorded. This distribution column describes the distribution of the data under which the AI/ML constituent is expected to operate [5]. The distribution of each parameter describes the assumed underlying distribution, from which the data were sampled independently [5]. These distributions are important to record, as the generalization capability of an ML model is approximated based upon the assumption that the out-of-sample data is sampled from the same distribution as the in-sample data [5]. This is crucial, as the generalization capability of an ML model is a probabilistic statement that is only valid if the assumption of a common underlying distribution holds [16]. Thus, distribution information is crucial to record, as it can later be used for ODD monitoring. Therefore, it is necessary to ensure that the ML model is exposed only to the data distributions on which it was trained, thereby enabling the AI/ML constituent to exhibit its intended behavior [5].

In addition to EASA requirements, it is proposed to add a column to record the sensor, enabling identification of the data source within the system. This information can be used in data management to verify data integrity, as knowledge of the sensor enables identification of potential errors introduced by the chosen sensor setup during data collection. Also, the inclusion of sensors in the ODD enables assessment of the sensor setup [24]. This assessment of the sensor setup and its characteristics enables the determination of whether the system-level sensor setup is sufficient for the defined AI/ML constituent ODD. Otherwise, an iteration and modification of the sensor setup at the system level might be required [24]. Importantly, not every ODD parameter can be measured directly by a sensor; some can only be inferred indirectly from other measurements. Therefore, these parameters may only need to be collected during data acquisition to ensure the specified conditions are captured. During operation, depending on the ML model, these parameters might not be directly used as input for the ML model in the training and in the inference during operation, and are only indirectly captured, for example, the weather conditions in an image. However, these attributes must still be included in the AI/ML constituent ODD and in the collected dataset to enable successful training of the ML model under the specified conditions. Lastly, based on sensor data and system-level safety assessments, it is possible, using methods such as ARP4761A [11], to further identify regions of potentially invalid or erroneous input data. Such an assessment can identify scenarios and data to test the ML model's behavior under these degraded conditions. This information can inform the definition of the system's ODD monitoring capability [5] and the development of mitigation strategies [6].

3.4.3. AI/ML Constituent ODD Specification

The previous two sections, Sections 3.4.1 and 3.4.2, defined the components for the ODD specification. As with the OD, a process is also needed to methodically develop an ODD specification for any use case or AI-based system. The proposed process is based on the *domain-agnostic and risk-based ODD definition approach* [61], which consists of the steps of initialization, refinement, and verification and validation of the AI/ML constituent ODD. The first step in the process is to initialize the AI/ML constituent ODD. This initialization is based on the functional decomposition, the OD, and the requirements allocated to the AI/ML constituent. It defines an ODD based on the system-level information. Once the initial definition of the ODD is complete, refine it to incorporate AI/ML constituent-specific information and concepts. This refinement includes details on the sensor setup and the AI/ML constituent architecture, which must be considered when defining the ODD. These considerations focus on incorporating the specifics required for data management and learning process management of the AI assurance process [5]. This is required, as the AI/ML constituent ODD must provide "a framework for the selection, collection, [and] preparation of the data" [5] that is used for the development of the ML inference model [5]. The last step in the process is the verification and validation of the AI/ML constituent ODD. This verification and validation are outlined in the anticipated MOC DA-07 [5], where the verification and validation consist of subject-matter experts reviewing the operating parameters for correctness and completeness.

The initialization of the AI/ML constituent ODD is based on the ConOps, OD, functional decomposition, and requirements allocated to the AI/ML constituent. Therefore, the artifacts from the AI trustworthiness analysis must be used to initialize the AI/ML constituent ODD [5]. For each OD attribute, it should be determined whether it is a relevant operating condition for the AI/ML constituent, based on the functions and requirements allocated to it. To guide this identification, each parameter should be classified into one of two groups: physical ranges or parameters that affect the data distribution, such as the behavior of dynamic elements. If an attribute is classified as a physical range, based on the allocated function and requirements of the AI/ML constituent, this range can then be adapted accordingly. The second option is to classify an attribute as influencing the data distribution. These attributes are important to recognize, as they will significantly affect how the individual samples are distributed in the dataset used to develop the ML inference model. Furthermore, during data collection, these attributes must be monitored to ensure that the conditions they imply are included in the dataset. Using this classification for each attribute, it can later be determined whether it is relevant to the AI/ML constituent and data management. Note that not all attributes will be distinguishable between the two groups; therefore, this classification will depend on the developers' subjective judgment. At the end of the AI/ML constituent ODD initialization, an initial ODD is created. The created ODD is a subset of the OD, since initialization relies only on the attributes and their ranges defined in the OD. This will likely change as the ODD is refined in the next stage of the process. Furthermore, this refinement can yield an AI/ML constituent ODD that is a superset of the system's ODD. Such a superset relationship can improve a model's performance and stability by training it on a broader range of available data [5].

Because the initialized AI/ML constituent ODD is based solely on the OD, which is insufficient for data management and learning process management [5], the next step is to refine the initialized ODD into a form suitable for these tasks. To achieve this, attribute projection, sensor characterization, domain-specific concept identification, and analysis of existing data sources are introduced. These steps aim to provide sufficient detail for the design and development of the ML inference model within the AI/ML constituent.

As discussed previously, the attributes of the initial ODD are based on the system-level OD. However, these attributes may be perceived differently by the AI/ML constituent than they are defined at the system level, largely due to the sensors selected in the system architecture. Therefore, the attributes and ranges of the initial AI/ML constituent ODD may need to be projected into the dimensional space that the AI/ML constituent can perceive. One necessary projection of attributes is the transformation of physical attributes. These transformations can be simple unit conversions or more complex operations, for example, when semantically defined attributes of the OD must be described using concrete physical sensor values. Another necessary transformation is converting geometric attributes into data-characteristic properties, where these properties depend on the available sensor [47]. Also, it may be necessary to apply these transformations in reverse to reconstruct geometric properties from image-level features. Applying these transformations to the attributes of the initial ODD should yield an ODD that aligns with the AI/ML constituent's perception and is usable in data management.

An important aspect of data management is collecting data to develop the ML inference model. It must be ensured that the data characteristics match those encountered by the ML inference model during its application [44,65]. One important aspect is the data properties driven by the sensors' characteristics and their configurations used in the system. These characteristics can include the sensor's orientation, type, and position, as well as properties determined by the sensor. In general, these properties can include the sensor's resolution or signal-to-noise ratio [66]. Such characteristics are important to describe, as they have been shown to negatively impact the performance of ML inference models [67]. The identification of these sensor characteristics can be based on the sensor specifications and how the sensor is installed in the overall system.

As previously explained, the OD describes the operating conditions of the AI-based system, while the AI/ML constituent ODD describes the operating conditions of only its corresponding AI/ML constituent. This discrepancy in definition implies additional or different conditions relative to the defined OD, as the AI/ML constituent may be influenced differently by those conditions. The additional conditions will largely depend on the AI/ML constituent's perception, e.g., its sensor setup. Furthermore, as the AI/ML constituent ODD must provide "a framework for the selection, collection, preparation of the data" [5], the ODD should also incorporate specific concepts that impact the ability of the ML inference model to learn its designated function and can affect the performance of the ML inference model in operation [47]. Due to the lack of robust methods for verifying ML inference outputs during operation, it is often difficult to detect erroneous outputs and identify their root causes. Therefore, identifying such concepts is important for designing the AI/ML constituent ODD. To identify such attributes for the AI/ML constituent ODD, other work has proposed methods to identify and classify necessary, supportive, irrelevant, and false-positive concepts [47]. This identification and classification help determine key aspects of the operating environment that can affect the ML model's ability to learn its designated function successfully. However, for these concepts, identifying the AI/ML constituent in the domain remains challenging. Because ML models, especially deep learning architectures, are designed to extract patterns and features from data, it is unclear to the developer which known or unknown patterns and features in the data are most significant for the model's ability to learn its designated function. Nevertheless, such an identification is important to ensure that the collected dataset accurately reflects the AI/ML system's later operating conditions. To identify attributes based on these concepts, an approach similar to the 6-layer model introduced in Section 3.2.3 can be used. Therefore, building on the same 6-layer model [33] used at the system level, an operating-condition model for the AI/ML constituent can be constructed. Importantly, the defined system-level model can serve as a basis, but must be extended based on the AI/ML constituent's perception. Furthermore, to identify all relevant model attributes that may affect the recorded sensor signals, the signal path should be mapped in the model [29]. Through thorough mapping, the different objects that interact with the signal can be discovered and described as an AI/ML constituent ODD attribute. Additionally, to accurately describe and model a use case, an ontology-based domain model can be developed [68]. Such a domain model represents the natural language domain knowledge in a graphical form [68]. The resulting graph consists of entities, relations, attributes, and values that describe the corresponding real-world concepts [68]. Based on the entities and attributes of the domain model, attributes in the AI/ML constituent ODD can be defined. The values associated with these entities and attributes in the domain model can then be used to define the value range of the corresponding AI/ML constituent ODD.

Depending on the use case, available data sources can be used to identify relevant aspects for the AI/ML constituent ODD [65]. Such data sources can include standards applicable to the use case, sensor specifications, or datasets from the same or a similar use case. Similar to the definition of the OD, standards can already specify a list of operating conditions under which the AI/ML constituent may operate. In the aviation sector, standards specify operational services and environmental definitions, as well as minimum aviation system performance requirements for systems used to provide certain functionalities. The information or requirements defined in these documents can be mapped to attributes in the AI/ML constituent ODD. In addition to these standards, sensor characteristics can be determined from the specifications provided for the sensors used in the AI-based system. The final source of AI/ML constituent ODD attributes considered are available datasets of the same or similar use cases [65]. The discovered datasets can be explored to identify additional attributes for the AI/ML constituent ODD. The exploration of a dataset involves identifying properties and their relationships within the dataset [69]. Importantly, such an exploration must be used to identify new attributes rather than merely to generate an AI/ML constituent ODD, as the available dataset may not fully represent the AI/ML constituent's operating environment.

3.4.4. AI/ML Constituent ODD Verification and Validation

The last step in the AI/ML constituent specification process is independent verification and validation of the AI/ML constituent ODD. As previously outlined, an insufficient AI/ML constituent ODD can pose a safety risk for the overall system [60]. The verification and validation of the correctness and completeness of the ODD must be done by subject-matter experts [5]. In addition to expert judgment, this verification should ensure that all relevant domain standards applicable to the system are considered [60]. Furthermore, additional datasets should be used in this verification and validation. Similar to the discovery of new attributes from data sources, these additional datasets can help ensure that the attributes and their values account for all relevant conditions [60].

3.5. Model Architecture and Input-Feature Selection

Following the definition of the AI/ML constituent ODD, the next step is to define the AI/ML constituent architecture, including preprocessing, the ML inference model, and post-processing required to perform the function allocated to the AI/ML constituent [5]. As the AI/ML constituent ODD is not the only influencing factor for the architecture, a mapping of the different influence factors is depicted in Figure 3. These influencing factors were identified based on the objectives outlined in EASA's concept paper [5].

As shown in Figure 3, the *Inputs* for the AI/ML constituent that are used in the ML model are influenced by the AI/ML constituent ODD, the type of data that is collected, and the architecture. The AI/ML constituent ODD determines the characteristics of the data to be captured, based on the required conditions and the AI/ML constituent ODD attributes and their ranges. However, the AI/ML constituent ODD does not specify the data type or format of the collected data. Therefore, the data can be structured or unstructured, which largely determines the types of inputs that are feasible for an ML model. While all these can provide the necessary information that might be described in the AI/ML constituent ODD, the different data types available for the same ODD allow different input features. One important step in defining the inputs is applying feature engineering methods to extract the most useful input features [70]. However, these feature engineering steps may not be necessary if an ML model architecture incorporates feature-extraction components. The architecture for the ML model is not directly dependent on the defined AI/ML constituent ODD; it is only influenced indirectly through the collected data types. In addition, the requirements [44] and the functions allocated to the AI/ML constituent have a major influence on the architecture. The selection of a suitable ML model architecture depends on the performance achieved by individual architectures across different experiments [71]. Most importantly, the components and the ML model must adhere to the specified requirements and the derived performance metrics.

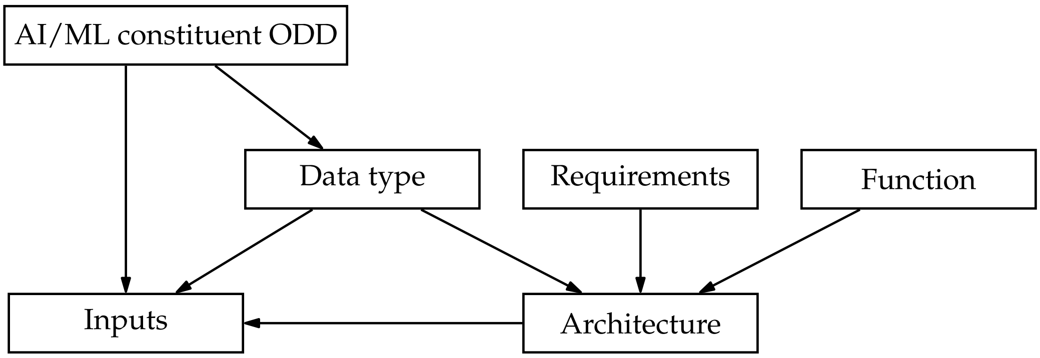

Figure 3. A mapping of the influence factors when designing an ML model.

4. Case Study

This section introduces the application of the proposed framework to the selected airborne collision avoidance system use case, aiming to validate whether it conforms to the considered EASA objectives, as shown in Table 1. For the use case, the vertical collision avoidance systems (VCAS) [72] and horizontal collision avoidance systems (HCAS) [73], both implemented in PYCASX [40,74], were selected. The PYCASX system is a Python-based implementation of HCAS and VCAS, inspired by ACAS Xa and ACAS Xu, and integrated with FlightGear [75–77]. This new system is proposed as a neural network-based solution to reduce the memory footprint of ACAS Xa/Xu to make its implementation on avionics hardware feasible [40,54]. It was selected as a representative use case because it builds on established research in the aviation AI community and provides a modular architecture that integrates easily into simulation environments for efficient testing. To reduce memory footprint, neural networks are used due to their excellent compression capabilities [53]. However, because neural network compression is not loss-free, the Safety Net concept is proposed [36,54,55], which uses neural networks and sparse lookup tables to achieve loss-free compression. As PYCASX utilizes neural networks, it is an AI-based system and must therefore conform to the considered objectives of the EASA concept paper [5]. For this reason, the framework introduced in the previous Section 3 is applied to PYCASX and its subsystems. Importantly, as PYCASX contains multiple subsystems that are not AI-based, the assumption is made that these subsystems will be built according to the relevant aviation guidelines and standards, such as DO-178C [2]. Therefore, in the following sections, these subsystems will be introduced only to the extent necessary to understand the VCAS and HCAS and the system's overall functionality.

4.1. Definition of a ConOps for pyCASX

The task of pyCASX is to provide partial ACAS capabilities to the pilots of an aircraft. For the ConOps definition, the following section walks through the individual steps defined in Section 3.1. In the given use case, this can include the two systems, VCAS and HCAS, based on ACAS Xa and ACAS Xu, respectively. Each of these systems or components might potentially be replaced by pyCASX. ACAS Xa is standardized by the DO-385 [75] and ACAS Xu by the DO-386 [76]. Importantly, ACAS Xu, designed to provide resolution advisories for pilots in the vertical and horizontal plane, is a superset of ACAS Xa. Thus, it is often sufficient to reason about ACAS Xu. In addition to these systems, further standards were identified that may apply to the ACAS use case, such as ED-271 [78] and ED-313 [79]. As previously introduced, the justification for the changes is the need to further reduce the memory footprint of ACAS Xu [54]. The description of the proposed AI-based system is based on the description provided by the initial implementation of pyCASX [40,74]. To summarize, pyCASX is designed to provide the pilots with last-resort measures to prevent mid-air collisions. The system is designed to operate in European airspace *type C*, where aircraft typically operate in the IFR or VFR mode. The airspace *type C* also entails that no geographic features or structures are present in the airspace. Both the ownship and all possible intruders must be equipped with an ADS-B system. Based on the aircraft's position and dynamics, the system can determine whether the ownship is on a collision course with an intruder. If an intruder is on a collision course, the system shall generate a resolution advisory for the pilot to prevent a potential collision. Resolution advisories can be on a vertical or horizontal plane. If multiple intruders are on a collision course, the system issues the strongest advisory to the pilot, provided no conflicting advisories were calculated for the intruders. Furthermore, no coordination between the ownship and intruder is assumed. The pilots must first determine whether the resolution advisory can be executed safely, then execute it.

4.2. Definition of an OD for pyCASX

Following the introduction of the system capabilities and the operational description in Section 4.1, the next step is to define the OD for the pyCASX system. To identify the elements of the OD, the environment was deconstructed using a layer model [33]. The model consists of six layers, based on the automotive 6-layer model [33], and adapted for the aviation use case to describe all elements of the airspace for en-route operations. Furthermore, this model was applied to the previously described ConOps to follow a system-oriented, risk-driven approach. Through this application, the OD attributes were systematically traced from the operational assumptions and operational scenarios of the ConOps to reduce the risk of omitting safety-relevant conditions for the system. The identified attributes and the resulting OD specification for the pyCASX system are shown in Table 2. For these attributes, values were initialized during initialization based on the ConOps and further refined using the identified standards applicable to pyCASX.

Table 2. The specified OD for PYCASX.

Top-Level Attribute	Sub-attribute	Qualifier	Attribute	Attribute Value	Unit
Scenery	Airspace	Include	Type	C	-
	Airspace	Include	Flight Rule	IFR, VFR	-
	Airspace	Include	Altitude	[10 000, 66 000]	ft
	Airspace	Include	Latitude	[−90, 90]	°
	Airspace	Include	Longitude	[−180, 180]	°
	Airspace	Include	Route Type	Free Route Airspace	-
	Airspace	Exclude	Geography	Any	-
	Airspace	Exclude	Structures	Any	-
Environment	Weather	Exclude	Adverse Conditions	Any	-
	Connectivity	Include	Satellite Positioning	GPS	-
	Connectivity	Include	Communication Type	ADS-B	-
	Connectivity	Include	Communication Range	≥ 20	NM
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-
	Intruder	Include	Maximum Agent Density	0.06	NM^{-2}
	Intruder	Include	Latitude	[−90, 90]	°
	Intruder	Include	Longitude	[−180, 180]	°
	Intruder	Include	Altitude	[10 000, 66 000]	ft
	Intruder	Include	Horizontal Airspeed	[0, 600]	kn
	Intruder	Include	Horizontal Acceleration	[−1.5, 1.5]	g
	Intruder	Include	Vertical Rate	[−5000, 5000]	$\mathrm{ft\,min}^{-1}$
	Intruder	Include	Vertical Rate Acceleration	$[-\frac{1}{3}, \frac{1}{3}]$	g
	Intruder	Include	Heading	[−180, 180]	°
	Intruder	Include	Communication Type	ADS-B	-
	Ownship	Include	Agent Type	Airplane	-
	Ownship	Include	Latitude	[−90, 90]	°
	Ownship	Include	Longitude	[−180, 180]	°
	Ownship	Include	Altitude	[10 000, 66 000]	ft
	Ownship	Include	Horizontal Airspeed	[0, 600]	kn
	Ownship	Include	Horizontal Acceleration	[−1.5, 1.5]	g
	Ownship	Include	Vertical Rate	[−5000, 5000]	$\mathrm{ft\,min}^{-1}$
	Ownship	Include	Vertical Rate Acceleration	$[-\frac{1}{3}, \frac{1}{3}]$	g
	Ownship	Include	Heading	[−180, 180]	°
	Ownship	Include	Vertical Rate Capability	≥ 2000	$\mathrm{ft\,min}^{-1}$
	Ownship	Include	Vertical Rate Acceleration Capability	$\geq \frac{1}{3}$	g
	Ownship	Include	Turn Rate Capability	≥ 3	$^\circ\,\mathrm{s}^{-1}$
	Ownship	Include	Turn Rate Acceleration Capability	≥ 1	$^\circ\,\mathrm{s}^{-2}$
	Ownship	Include	Pilot Type	Pilot, Remote Pilot	-

4.3. Functional Decomposition of PYCASX

Based on the introduced ConOps, a functional decomposition of the PYCASX was conducted. The first primary function identified was detecting other aircraft and their positions. This function is split into three subfunctions that are necessary to sense the position, height, and velocity of the ownship and surrounding intruders. The second main function determines if a collision with the intruder is imminent. This is further split into subfunctions to preprocess data gathered by the first function, to calculate and select the optimal advisory, and to resolve conflicts among advisories when multiple are generated for multiple intruders. The third main function is the interface to the aircraft avionics. This main function comprises multiple subfunctions that display and announce the selected advisories to pilots. Significantly, only the subfunctions in the second main function, which determine the best possible advisory, are AI/ML-based, while all other functions are traditional software and hardware items. Additionally, a preliminary system architecture for PYCASX is defined to implement the specified functions, as shown in Figure 4. The functions for sensing the intruder's position and ownship are implemented by *ADS-B* and *GPS* in the presented architecture. The determination whether a collision is imminent is realized by the *CPA calculation*, *HCAS*, *VCAS*, and the *Multi-intruder advisory selection* components. Here *HCAS* generates advisories in the horizontal plane, while *VCAS* does so in the vertical plane. The *Display & Speaker* component is responsible for implementing the function that alerts pilots to the determined advisory.

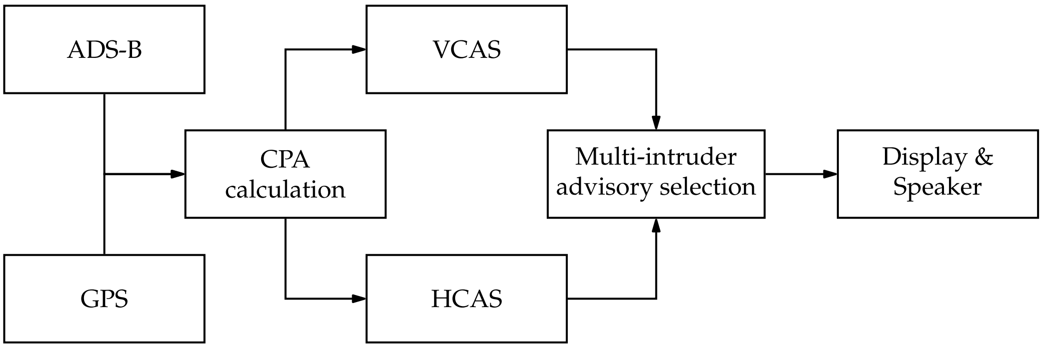

Figure 4. The preliminary architecture of PYCASX [40,74].

4.4. Definition of an AI/ML Constituent ODD for VCAS and HCAS

Following the previous steps for the PYCASX system, the next step in developing the AI/ML constituents is to define an ODD for each constituent, i.e., an AI/ML constituent ODD for VCAS and an AI/ML constituent ODD for HCAS. This is necessary because VCAS provides advisories only in the vertical plane, and HCAS only in the horizontal plane. Therefore, these AI/ML constituents are created based on different datasets, and each dataset requires an individually defined AI/ML constituent ODD. Importantly, as both AI/ML constituents are part of the PYCASX system, the same upstream artifacts of the ConOps, system-level OD, and functional decomposition are used to derive the two AI/ML constituent ODDs, ensuring system-level traceability.

4.4.1. The AI/ML Constituent ODD for VCAS

Applying the methodology described in Section 3.4.3, the previously defined ConOps, OD, and functional decomposition of PYCASX are used in the definition of the AI/ML constituent ODD. Because VCAS provides only vertical advisories, the AI/ML constituent ODD is defined only for data in this plane.

Using the defined OD from Section 4.2, the first step is the identification of the individual parameters of the OD that must be part of the AI/ML constituent ODD. For this, as described in Section 3.4.3, all attributes were classified as attributes of a physical range or as behavior-determining attributes. Using this classification and the allocated function to VCAS for each attribute, it is determined if it is relevant for the AI/ML constituent ODD. For example, the OD attributes of the airspace, *altitude* and *coordinates* are both classified as physical ranges. Because VCAS can resolve conflicts regardless of airspace dimension, these are considered irrelevant to the VCAS ODD. Similar to the *type*, the *flight rule* and *route type* are both elements that determine the behavior of aircraft and are, therefore, relevant for the VCAS ODD. Other OD attributes, such as *geography* and *structures*, are determined to be irrelevant to the VCAS ODD, as the detection of these conflicts is handled by other systems, for example, the ground proximity warning system [80]. The same steps were done for the attributes of *environment* and *dynamic elements*. Based on these results, an initial ODD for VCAS was defined and subsequently refined in the following steps.

Following the definition of the initial ODD for VCAS, the next step is to project attributes into the VCAS system's perception. In this case, only the *altitude* attribute is affected. With VCAS, it is not the absolute altitude of the ownship and the intruder that matters, but the difference in altitude. Therefore, for data management, it will be important to collect data with sufficient samples across all possible relative altitudes between the ownship and the intruder. The acquisition of data to satisfy the defined ODD attributes is typically achieved through a combination of empirical real-world measurements and synthetic data generated through simulation-based testing to cover safety-critical edge cases that are often underrepresented or too hazardous to capture in the real world. The maximum relative altitude (Δh_{max}) between ownship and intruder is given by the maximum vertical rate ($\dot{h}_{\text{max}}$) and the maximum time to the closest point of approach (τ) and therefore can be calculated based on the defined OD values.

The next step for the VCAS ODD is to identify sensor characteristics. The only sensors used in PYCASX are GPS and ADS-B. ADS-B is the broadcast of information about the aircraft's position, speed, and altitude based on data from the global navigation satellite system (GNSS) [81]. For both the ownship and intruder, GNSS is assumed to utilize GPS; therefore, both are affected by the same sensor characteristics. Because GPS directly provides the required information, the only additional sensor characteristic introduced is GPS measurement inaccuracy. However, because GPS has high accuracy [82] and at large heights only uncommon causes, such as solar storms [83], can cause larger inaccuracies, these inaccuracies are assumed to be zero.

To identify additional domain-specific concepts for VCAS, an ontology-based domain model was created for an encounter between an ownship and a single intruder. This ontology-based domain model was built according to the rules outlined in Section 3.4.3. As shown in Figure 5, the model contains two entities, the *Intruder* and the *Ownship*. Both entities share the three attributes *Velocity*, *GNSS*, and *Position*. For the ownship one special attribute was identified, namely the *Pilot*, characterized by the attribute *Reaction Time*. This attribute specifies the time required for the pilot from the advisory announcement to the start of advisory execution. Only when the delay is accounted for can the system generate advisories earlier, still allowing sufficient time for safe avoidance [84]. In this work, the reaction time is assumed to be 0 s, in line with previous research [40,53,54,85,86].

For VCAS, a dataset used to build the original version of VCAS [53] is available. In addition to the VCAS advisories, the dataset contains only four attributes for the encounter data. These attributes were already identified in previous steps; therefore, this dataset yielded no additional attributes for the VCAS ODD.

The final AI/ML constituent ODD for VCAS is shown in Table 3. It includes three attributes under the top-level attribute *Scenery*, eleven under *Dynamic Elements*, and two under *Operating Parameters*. Compared to the PYCASX OD, the defined VCAS ODD contains fewer attributes. These fewer attributes result from the allocated function and requirements for the AI/ML constituent, as it only needs to cover a smaller subset of conditions than the system-level conditions. The three attributes for the *Scenery* influence the behavior of the ownship and intruder, ensuring that these distinct behaviors are captured in the dataset. Under the *Dynamic Elements*, most of the identified attributes are physical ranges that determine the allowed distances and rates of the intruder and the ownship. A specialty for the ownship is the inclusion of attributes that require certain performance criteria for the ownship itself. These attributes are already included in OD but are also relevant for the AI/ML constituent ODD to ensure that the ownship conforms to the requirements needed to execute the advisories as expected. The *Distribution* was determined only for attributes that are constant, i.e., whose range includes a single value. For all other attributes, no distribution was specified, as their distributions depend on the collected data. Therefore, this column can only be completed once the data has been collected and used to build the final AI/ML constituent.

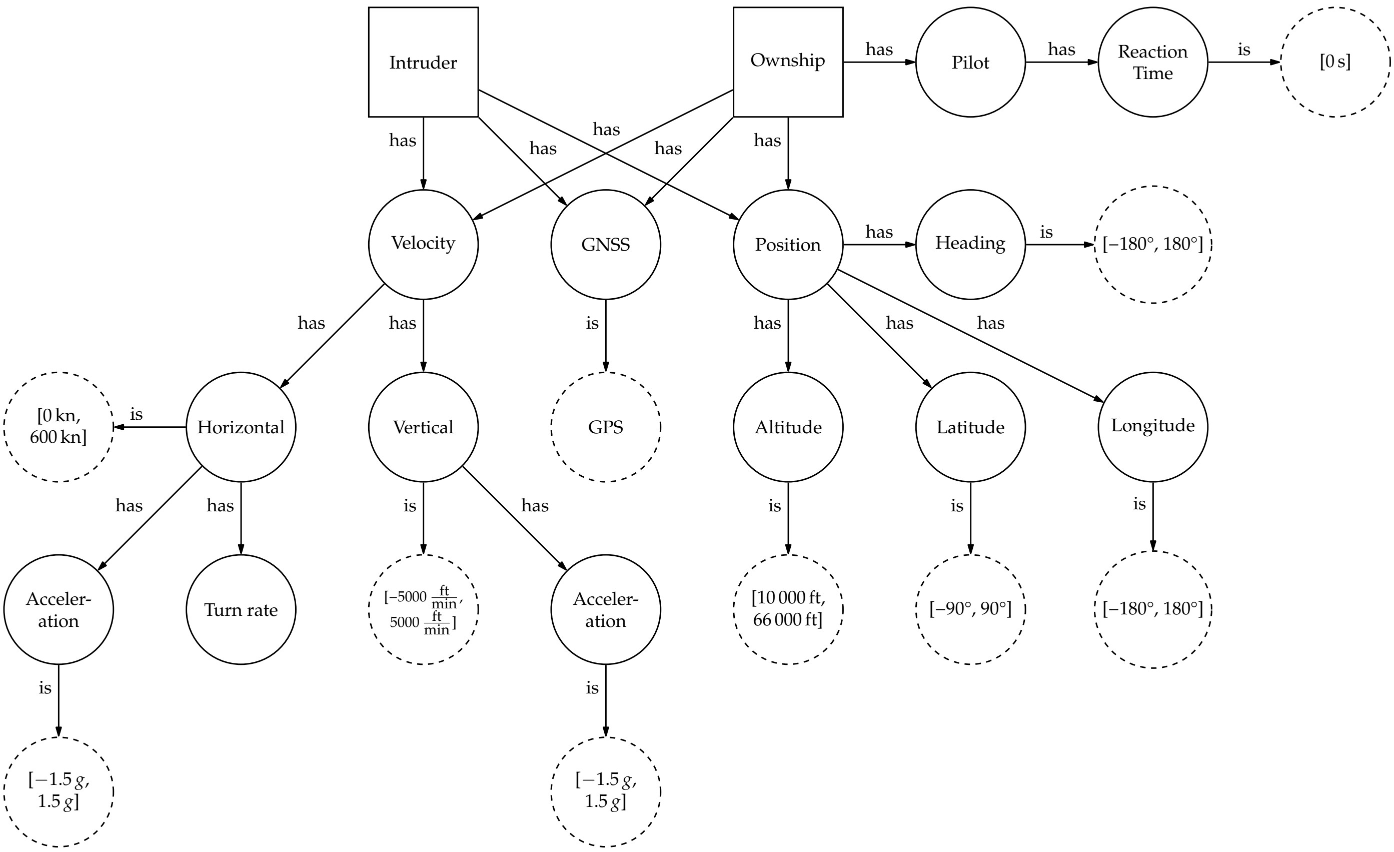

Figure 5. The complete ontology-based domain model for an encounter between the ownship and an intruder. Entities are depicted with a square, attributes by circles, values by dashed circles, and relations by arrows.

4.4.2. The AI/ML Constituent ODD for HCAS

As HCAS is similar to VCAS, the following section introduces the AI/ML constituent ODD for HCAS, but only in a shortened form that highlights differences from the VCAS ODD definition. The initialization of the ODD for HCAS follows a similar approach to VCAS, with the only difference that, for the *Dynamic Elements*, all attributes in the vertical plane are discarded, and only those in the horizontal plane are retained. Because HCAS also relies on the same sensors as VCAS, the same additional GPS accuracy attributes were identified for the *Operating Parameters*. The identification of domain-specific concepts for HCAS was carried out using the same ontology-based domain model as for VCAS. Similar to VCAS, this model provided only the pilot's *Reaction Time* as an additional attribute. The finalized AI/ML constituent ODD for HCAS is shown in Table 4. As already described, the major difference between the VCAS ODD and the defined HCAS ODD is that the encounters are on the horizontal plane. Therefore, the attributes of the *Scenery*, the *Environment*, and the *Operating Parameters* are the same as in the defined VCAS ODD. Only for the top-level attribute of *Dynamic Elements*, new attributes were introduced to describe the encounter geometry, and attributes of the vertical plane were discarded. This also includes attributes that ensure that the ownship executes these advisories in the dataset as specified. Therefore, the attributes of the *Turn Rate Capability* and the *Turn Rate Acceleration Capability* are included in the HCAS ODD.

Table 3. AI/ML constituent ODD for VCAS.

Top-Level Attribute	Sub-Attribute	Qualifier	Attribute	Attribute Value	Unit	Distribution	Source
Scenery	Airspace	Include	Type	C	-	Constant	-
	Airspace	Include	Flight Rule	IFR, VFR	-	-	-
	Airspace	Include	Route Type	Free Route Airspace	-	Constant	-
Environment	-	-	-	-	-	-	-
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-	Constant	ADS-B
	Intruder	Include	Vertical Rate	[−5000, 5000]	$\mathrm{ft\,min^{-1}}$	-	ADS-B
	Intruder	Include	Vertical Rate Acceleration	$[-\frac{1}{3}, \frac{1}{3}]$	g	-	-
	Intruder	Include	Relative Altitude to Ownship	[−10 000, 10 000]	ft	-	-
	Intruder	Include	Time until loss of separation	[0, 60]	s	-	-
	Ownship	Include	Agent Type	Airplane	-	Constant	-
	Ownship	Include	Vertical Rate	[−5000, 5000]	$\mathrm{ft\,min^{-1}}$	-	GPS
	Ownship	Include	Vertical Rate Acceleration	$[-\frac{1}{3}, \frac{1}{3}]$	g	-	GPS
	Ownship	Include	Vertical Rate Capability	≥2000	$\mathrm{ft\,min^{-1}}$	-	-
	Ownship	Include	Vertical Rate Acceleration Capability	$\geq \frac{1}{3}$	g	-	-
	Ownship	Include	Pilot reaction time	[0]	s	Constant	-
Operating Parameters	Ownship	Include	GPS Inaccuracy	None	-	Constant	GPS
	Intruder	Include	GPS Inaccuracy	None	-	Constant	GPS

4.5. Framework Validation

The goal of the proposed framework is to enable developers of AI-based systems to meet the objectives set by EASA. For the framework, a subset of all of EASA's objectives was selected. These selected objectives then imposed conditions that must be met by the individual steps in the framework. They were later verified in Table 5 by the application to the pyCASX use case. The framework fulfilled 72.2 % of the considered objective requirements. No requirements were unfulfilled, and 27.7 % of the considered objective requirements were only partially fulfilled. Those were not fully met, as either the methodology yielded results only for the use case that indirectly covered the requirements, or the results depended on later steps of the W-shaped learning assurance process and therefore could only be achieved at the end of development. Furthermore, as shown in the previous sections, the framework extended and clarified concepts from the EASA concept paper, such as the OD and the AI/ML constituent ODD. Importantly, as shown in the steps for defining the ODD for VCAS and HCAS, an approach was demonstrated that maps from the system level to the AI/ML constituent level. Thus, applying the framework to the pyCASX use case confirmed the benefits of the introduced formalisms and developer-required processes, thereby completing the necessary validation of the framework.

Table 4. AI/ML constituent ODD for HCAS.

Top-Level Attribute	Sub-Attribute	Qualifier	Attribute	Attribute Value	Unit	Distribution	Source
Scenery	Airspace	Include	Type	C	-	Constant	-
	Airspace	Include	Flight Rule	IFR, VFR	-	-	-
	Airspace	Include	Route Type	Free Route Airspace	-	Constant	-
Environment	-	-	-	-	-	-	-
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-	Constant	ADS-B
	Intruder	Include	Horizontal Airspeed	[0, 600]	kn	-	ADS-B
	Intruder	Include	Horizontal Acceleration	[−1.5, 1.5]	g	-	-
	Intruder	Include	Relative Angle to Ownship	[−180, 180]	°	-	-
	Intruder	Include	Time until loss of separation	[0, 60]	s	-	-
	Ownship	Include	Agent Type	Airplane	-	Constant	-
	Ownship	Include	Horizontal Airspeed	[0, 600]	kn	-	GPS
	Ownship	Include	Horizontal Acceleration	[−1.5, 1.5]	g	-	GPS
	Ownship	Include	Distance to Intruder	[0, 122 000]	ft	-	-
	Ownship	Include	Angle to Intruder	[−180, 180]	°	-	-
	Ownship	Include	Turn Rate Capability	≥3	$^\circ\,\mathrm{s}^{-1}$	-	-
	Ownship	Include	Turn Rate Acceleration Capability	≥1	$^\circ\,\mathrm{s}^{-2}$	-	-
	Ownship	Include	Pilot reaction time	[0]	s	Constant	-
Operating Parameters	Ownship	Include	GPS Inaccuracy	None	-	Constant	GPS
	Intruder	Include	GPS Inaccuracy	None	-	Constant	GPS

Table 5. Identified and considered requirements of EASA's concept paper [5], which the methodology application must fulfill.

Objective	Requirement	Fulfilled
CO-01	Identification of end users	Partially
CO-02	Goals of the end users	Partially
	High-level tasks of the end users	Completely
CO-04	Operational scenarios	Completely
	Task allocation in the operational scenarios	Completely
	Capturing of operating conditions	Completely
CO-06	Functional decomposition of the system	Completely
	Function allocation in the system architecture	Completely
	Classification of AI/ML items	Completely
DA-03	Set of parameters for the AI/ML constituent	Partially
	Traced parameters to the OD	Completely

5. Discussion

In the previous chapter, the feasibility of the proposed framework for meeting the requirements associated with the considered objectives (see Table 1) was demonstrated. With this novel framework, most objectives for the selected use case of PYCASX and its corresponding AI/ML constituents, VCAS and HCAS, were achieved. The failure to meet all objectives was due to the proposed framework covering only a subset of the steps in developing an AI-based system. For example, for the AI/ML constituent ODD, not for every attribute, a distribution was determined, as these distributions depend on the collected datasets. These datasets can only be determined once the AI/ML constituent is fully developed in accordance with EASA's guidelines. Nevertheless, although it did not fully meet all requirements for applying the methodology to the use case, it demonstrated the advantage of consolidating diverse approaches and concepts into a unified AI Engineering approach. Importantly, the main difference of the suggested framework is the explicit separation between the system-level OD and the data-centric AI/ML constituent ODD, as required by EASA's guidelines [5]. This contrasts with existing approaches in the aviation and automotive domains, which primarily focus on scenario-based or system-level ODD descriptions.

Within the framework for specifying the OD and AI/ML constituent ODD, a tabular format was chosen. While this has the advantage of being easily readable by different stakeholders, it lacks an abstract syntax that formalizes the OD and ODD metamodels. This could be improved by using a modeling language, such as SysML v2, to define the abstract and concrete syntax for the OD and AI/ML constituent ODD. Furthermore, such a structured approach to specifying an OD and an AI/ML constituent ODD enables its use in system verification. Based on the developed specification language, intermediate translation steps can be defined to enable translation into a formal specification language [87]. This transformed OD or AI/ML constituent ODD representation enables the use of standard verification tools, such as satisfiability modulo theories (SMT) solvers, to check the specification for consistency and to support situation verification [87]. While the tabular format allows stakeholders without formal methods expertise to understand the OD or ODD specification, the required intermediate translation steps introduce an additional layer of complexity in verifying AI-based systems.

Two aspects not covered by this tabular format are conditional statements with multiple parameters and temporal aspects, which must be described in an OD or ODD specification. For example, depending on the flight rules permitted in the airspace, the allowable ranges of other parameters may be restricted. Consequently, the proposed format can represent only the union of all parameter ranges. This limitation should be addressed in future extensions of the format, particularly for AI-based systems operating in highly complex environments with a high level of autonomy, where more expressive constructs may be required to specify the OD and the AI/ML constituent ODD.

In addition, beyond EASA compliance, the proposed ODD methodology serves as a technical specification for subsequent ML development phases. By defining not only ranges but also distributions and attribute sources, the ODD serves as a sampling blueprint for data collection. For instance, in the HCAS use case, specifying the distribution of the *Turn Rate Capability* ensures that the training dataset is not biased toward linear trajectories, but instead contains sufficient edge cases of high-maneuverability encounters. This prevents data gaps that could lead to model overfitting or failure on corner cases. Furthermore, during model training, these ODD attributes constrain data augmentation and synthetic data generation, ensuring that generated scenarios remain within the operationally relevant envelope defined at the system level.

The advantage of the introduced framework is evident when comparing the results of the defined AI/ML constituent ODD with other AI/ML constituent ODDs specified for the same use case. One example of an ODD, defined by the MLEAP consortium [18] for the same use case of the airborne collision avoidance system, is shown in Figure 6. The specified ACAS Xu ODD only contains six attributes and their ranges. The HCAS ODD of this work is equivalent to the shown ACAS Xu ODD. Compared with the ODD specified in this work, the MLEAP ACAS Xu ODD lacks attribute classifications, units, qualifiers, attribute sources, and attribute distributions. The latter is significant, as it is a requirement for an AI/ML constituent ODD per EASA [5]. Therefore, the specified ACAS Xu ODD, as shown in Figure 6, does not fulfill the requirements of EASA. Furthermore, while the specified HCAS ODD contains 16 attributes, the MLEAP ACAS Xu ODD contains only 6. The six attributes are all contained in the HCAS ODD, and the major difference in attributes is the attributes that determine the behavior of the ownship and intruder, such as the *Turn Rate Capability*, see Table 4. However, it is necessary to include these attributes in the AI/ML component ODD, as they determine behavioral characteristics that can alter the optimal advisories across scenarios. Therefore, excluding these attributes will result in an insufficient AI/ML constituent ODD that does not accurately describe the AI/ML constituent's operating conditions. A sufficient level of detail is important, as only if the AI/ML constituent ODD accurately describes the conditions the AI/ML constituent can encounter can a safety argument be built on the AI/ML constituent ODD for the system, allowing for later certification of the system [60]. The missing parameters in the MLEAP ACAS Xu ODD also pose a problem for AI/ML constituent ODD monitoring, as they are not included in the AI/ML constituent ODD and therefore will not be monitored. This is problematic because it cannot be determined whether the AI/ML constituent was designed to operate in these scenarios. Furthermore, as the AI/ML constituent ODD is also the framework for the selection, collection, and preparation of the data used to develop the ML inference model, the completeness of the data also depends on the defined AI/ML constituent ODD [5]. For example, the parameter of the *Agent Type* is included in the HCAS ODD but is missing in the ACAS Xu ODD. This allows for all types of agents, such as rotorcraft, to be included in the dataset, while the HCAS ODD is limited to airplanes. This again underscores the need for an adequate AI/ML constituent ODD specification process.

$$ODD_{\text{ACAS Xu}} = \begin{cases} \tau \in [0, 101] \\ \theta \in [-3.14, 3.14] \\ v_{\text{int}} \in [0, 1200] \\ v_{\text{own}} \in [50, 1200] \\ \psi \in [-3.14, 3.14] \\ \rho \in [499, 185\,318] \end{cases}$$

Figure 6. The ACAS Xu ODD defined in the MLEAP report [18]. From top to bottom, the individual attributes describe the time to loss of separation in seconds, the relative angle to the intruder, the speed of the intruder in feet per minute, the speed of the ownship in feet per minute, the intruder heading relative to the onwship, and the distance of the intruder to the ownship in feet.

As introduced in Section 2, the EASA guidance for level 1 and 2 machine learning applications was released only in 2024, and guidance for level 3 is expected in the coming years. Therefore, ambiguities and gaps remain in the currently available guidance [88]. For example, other studies have identified issues or areas for improvement in certain AI assurance objectives [88]. The primary challenge encountered in developing the methodology for this work was the ambiguity surrounding some of the objectives under consideration. One example is the EASA definition of the OD; it is defined similarly to the AI/ML constituent ODD, in that it should describe the operating conditions at the system level [5]. However, their definition is unclear regarding whether the concept of OD is based solely on existing practices in the aviation sector or also on practices in other industries, such as the automotive sector. In this work, the latter interpretation was chosen. Also, for the AI/ML constituent ODD, it is unclear whether EASA permits specification of the AI/ML constituent ODD based on collected data, or whether it relies solely on a top-down allocation of requirements and functions. Again, the latter interpretation was chosen.

6. Conclusions

This work introduced a concrete, step-by-step framework that bridges AI trustworthiness analysis and learning assurance through explicitly defined development artifacts. Furthermore, the framework addresses the specification of operating conditions for AI-based systems in the context of potential future EASA regulations. The framework includes formats for defining OD, functional decomposition, and the AI/ML constituent ODD. In particular, for the AI/ML constituent of the ODD, additions to this format were made to incorporate data-specific conditions required by EASA. In addition to the required concept formats, processes were introduced to apply these concepts to concrete use cases. Importantly, a process was introduced to derive the AI/ML constituent ODD from the OD and the system description. Furthermore, a mapping of the individual factors influencing the AI/ML constituent and the ML model's inputs and architecture was created. The framework was applied to an airborne collision-avoidance use case to validate its capabilities and to introduce a neural network-based compression approach. Based on the use case, a ConOps, an OD, a functional decomposition, and an AI/ML constituent ODD were specified for the PYCASX system in accordance with the defined methodology. It was verified that most of the requirements of EASA's objective were satisfied for the chosen use case.

As discussed in Section 5, this work provided a baseline for defining the ConOps, OD, and AI/ML constituent ODD, but there remain gaps and areas for improvement. First, the currently available processes and guidance are largely conceptual and require further refinement and concretization. This concretization can take the form of developing a tool and an abstract syntax for defining the OD and AI/ML constituent ODD. Secondly, the current processes and the conducted validation considered only the specification of the system's operating conditions; therefore, the next step is to define data management processes based on the defined AI/ML constituent ODD. Lastly, as the introduced processes currently rely solely on the available EASA guidance for level 1 and 2 machine learning applications, adaptations may be necessary in light of the forthcoming level 3 guidance. Furthermore, the proposed processes should be applied to additional safety-critical use cases [89–91] to validate their usefulness beyond the chosen use case and its domain. In particular, a more data-driven use case, especially one not yet governed by existing standards, could further test and refine the introduced framework. Moreover, a level-3 use case could be selected to identify gaps in current processes. Such an application could also provide feedback to EASA in the development of its Level 3 machine learning guidance.

Author Contributions: Conceptualization, F.W., J.M.C. and T.S.; methodology, F.W., J.M.C. and T.S.; validation, F.W.; investigation, F.W.; resources, J.M.C.; visualization, F.W.; writing—original draft preparation, F.W.; writing—review and editing, J.M.C., T.S., E.H. and S.H.; supervision, S.H. and F.K.; project administration, J.M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: For this work, no new data were created.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACAS	Airborne Collision Avoidance System
ADS-B	Automatic Dependent Surveillance-Broadcast
AI	Artificial Intelligence
AIR	Aerospace Information Report
ARP	Aerospace Recommended Practice
ConOps	Concept of Operations
CPA	Closest Point of Approach
DO	Document
EASA	European Union Aviation Safety Agency
ED	EUROCAE Document
EUROCAE	European Organization for Civil Aviation Equipment
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HCAS	Horizontal Collision Avoidance System
IFR	Instrument Flight Rules
ML	Machine Learning
NM	Nautical Mile
OD	Operational Domain
ODD	Operational Design Domain
SAE	SAE International
SysML	Systems Modeling Language
VCAS	Vertical Collision Avoidance System
VFR	Visual Flight Rules

References

1. S-18 Aircraft and Sys Dev and Safety Assessment Committee. In *Guidelines for Development of Civil Aircraft and Systems*; SAE International: Machesney Park, IL, USA,2010. https://doi.org/10.4271/ARP4754A.
2. RTCA, Inc. *Software Considerations in Airborne Systems and Equipment Certification*; Techreport DO-178C; GlobalSpec: Washington, DC, USA, 2012.
3. RTCA, Inc. *DO-200B - Standards for Processing Aeronautical Data*; Technical Report; GlobalSpec: Washington, DC, USA, 2015.
4. G-34 Artificial Intelligence in Aviation. In *Artificial Intelligence in Aeronautical Systems: Statement of Concerns*; SAE International: Machesney Park, IL, USA, 2021. https://doi.org/10.4271/AIR6988.
5. European Union Aviation Safety Agency (EASA). *EASA Concept Paper: Guidance for Level 1 & 2 Machine Learning Applications*; Techreport; European Union Aviation Safety Agency (EASA): Cologne, Germany, 2024.
6. Kaakai, F.; Adibhatla, S.; Pai, G.; Escorihuela, E. Data-centric Operational Design Domain Characterization for Machine Learning-based Aeronautical Products. In *Proceedings of the International Conference on Computer Safety, Reliability, and Security, Toulouse, France, 20–22 September 2023*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 227–242.
7. Mamalet, F.; Jenn, E.; Flandin, G.; Delseny, H.; Gabreau, C.; Gauffriau, A.; Beaudouin, B.; Ponsolle, L.; Alecu, L.; Bonnin, H.; et al. *White Paper Machine Learning in Certified Systems*; Research Report, IRT Saint Exupéry; ANITI: Noida, India, 2021.
8. European Union Aviation Safety Agency (EASA). *Artificial Intelligence Roadmap 2.0*; Techreport; European Union Aviation Safety Agency (EASA): Cologne, Germany, 2023.
9. European Union Aviation Safety Agency. The Agency. Available online: https://www.easa.europa.eu/en/the-agency/the-agency (accessed on 15 August 2024).
10. Durand, J.G.; Dubois, A.; Moss, R.J. Formal and Practical Elements for the Certification of Machine Learning Systems. In Proceedings of the 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), Barcelona, Spain, 1–5 October 2023; pp. 1–10. https://doi.org/10.1109/DASC58513.2023.10311201.
11. S-18 Aircraft and Sys Dev and Safety Assessment Committee. In *Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment*; SAE International: Machesney Park, IL, USA, 2023. https://doi.org/10.4271/ARP4761A.
12. European Organization for Civil Aviation Equipment (EUROCAE). *Software Considerations in Airborne SYSTEMS and Equipment Certification*; Tech Report ED-12C; European Organization for Civil Aviation Equipment (EUROCAE): Malakoff, France, 2012.
13. RTCA, Inc. *Design Assurance Guidance for Airborne Electronic Hardware*; Tech Report; GlobalSpec: Washington, DC, USA, 2000.
14. European Organization for Civil Aviation Equipment (EUROCAE). *Design Assurance Guidance for Airborne Electronic Hardware*; Tech Report ED-80; European Organization for Civil Aviation Equipment (EUROCAE): Malakoff, France, 2000.

15. Federal Aviation Administration. *Roadmap for Artificial Intelligence Safety Assurance*; Technical Report; Federal Aviation Administration: Washington, DC, USA, 2024.
16. EASA and Daedalean. *Concepts of Design Assurance for Neural Networks (CoDANN)*; Resreport; European Union Aviation Safety Agency (EASA): Cologne, Germany, 2020.
17. EASA and Daedalean. *Concepts of Design Assurance for Neural Networks (CoDANN) II with Appendix B*; Resreport; European Union Aviation Safety Agency (EASA): Cologne, Germany, 2024.
18. MLEAP Consortium. *EASA Research—Machine Learning Application Approval (MLEAP) Final Report*; Technical Report 1; European Union Aviation Safety Agency (EASA): Cologne, Germany, 2024.
19. Luettig, B.; Akhiat, Y.; Daw, Z. ML meets aerospace: Challenges of certifying airborne AI. *Front. Aerosp. Eng.* **2024**, *3*, 1475139. https://doi.org/10.3389/fpace.2024.1475139.
20. Bello, H.; Geißler, D.; Ray, L.; Müller-Divéky, S.; Müller, P.; Kittrell, S.; Liu, M.; Zhou, B.; Lukowicz, P. Towards certifiable AI in aviation: Landscape, challenges, and opportunities. *arXiv* **2024**, arXiv:2409.08666.
21. *PAS 1883:2020*; Operational Design Domain (ODD) Taxonomy for an Automated Driving System (ADS)—Specification. BSI Standards Limited: London, UK, 2020.
22. *ISO 34503*; Road Vehicles—Test Scenarios for Automated Driving Systems—Specification for Operational Design Domain. Beuth Verlag: Berlin, Germany, 2023.
23. On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*; SAE International: Machesney Park, IL, USA, 2021. https://doi.org/10.4271/J3016_202104.
24. Eichenseer, F.; Sarkar, S.; Shakeri, A. A Systematic Methodology for Specifying the Operational Design Domain of Automated Vehicles. In Proceedings of the 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW), Tsukuba, Japan, 28–31 October 2024; pp. 13–18. https://doi.org/10.1109/ISSREW63542.2024.00040.
25. Thorn, E.; Kimmel, S.C.; Chaka, M. *A Framework for Automated Driving System Testable Cases and Scenarios*; Technical Report; United States Department of Transportation. National Highway Traffic Safety Administration: Washington, DC, USA, 2018.
26. Hallerbach, S. Simulation-Based Testing of Cooperative and Automated Vehicles. Ph.D. Thesis, Carl von Ossietzky Universität, Oldenburg, Germany, 2020.
27. Pegasus Research Project. *The Pegasus Method*; Pegasus Research Project: Köflach, Austria, 2019.
28. Fjørtoft, K.; Rødseth, Ø. Using the operational envelope to make autonomous ships safer. In Proceedings of the The 30th European Safety and Reliability Conference, Venice, Italy, 1–5 November 2020.
29. Picard, S.; Chapdelaine, C.; Cappi, C.; Gardes, L.; Jenn, E.; Lefevre, B.; Soumarmon, T. Ensuring Dataset Quality for Machine Learning Certification. In Proceedings of the 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Coimbra, Portugal, 12–15 October 2020; pp. 275–282. https://doi.org/10.1109/ISSREW51248.2020.00085.
30. Czarnecki, K. *Operational World Model Ontology for Automated Driving Systems–Part 1: Road Structure*; Technical Report; Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo: Waterloo, ON, Canada, 2018. https://doi.org/10.13140/RG.2.2.15521.30568.
31. Czarnecki, K. *Operational World Model Ontology for Automated Driving Systems–Part 2: Road Users, Animals, Other Obstacles, and Environmental Conditions*; Technical Report; Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo, Waterloo, ON, Canada, 2018. https://doi.org/10.13140/RG.2.2.11327.00165.
32. Mendiboure, L.; Benzagouta, M.L.; Gruyer, D.; Sylla, T.; Adedjouma, M.; Hedhli, A. Operational Design Domain for Automated Driving Systems: Taxonomy Definition and Application. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 4–7 June 2023; pp. 1–6. https://doi.org/10.1109/IV55152.2023.10186765.
33. Scholtes, M.; Westhofen, L.; Turner, L.R.; Lotto, K.; Schuldes, M.; Weber, H.; Wagener, N.; Neurohr, C.; Bollmann, M.; Körtke, F.; et al. 6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment. *IEEE Access* **2021**, *9*, 59131–59147. https://doi.org/10.1109/ACCESS.2021.3072739.
34. Stefani, T.; Jameel, M.; Gerdes, I.; Hunger, R.; Bruder, C.; Hoemann, E.; Christensen, J.M.; Girija, A.A.; Köster, F.; Krüger, T.; et al. Towards an Operational Design Domain for Safe Human-AI Teaming in the Field of AI-Based Air Traffic Controller Operations. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 29 September–3 October 2024; pp. 1–10. https://doi.org/10.1109/dasc62030.2024.10749684.
35. Berro, C.; Deligiannaki, F.; Stefani, T.; Christensen, J.M.; Gerdes, I.; Köster, F.; Hallerbach, S.; Raulf, A. Leveraging Large Language Models as an Interface to Conflict Resolution for Human-AI Alignment in Air Traffic Control. In Proceedings of the 2025 AIAA DATC/IEEE 44th Digital Avionics Systems Conference (DASC), Montreal, QC, Canada, 14–18 September 2025; pp. 1–10.
36. Gabreau, C.; Gauffriau, A.; Grancey, F.D.; Ginestet, J.B.; Pagetti, C. Toward the certification of safety-related systems using ML techniques: The ACAS-Xu experience. In Proceedings of the 11th European Congress on Embedded Real Time Software and Systems (ERTS 2022), Toulouse, France, 1–2 June 2022; pp. 1–11.
37. Stefani, T.; Anilkumar Girija, A.; Mut, R.; Hallerbach, S.; Krüger, T. From the Concept of Operations towards an Operational Design Domain for safe AI in Aviation. In Proceedings of the DLRK 2023, Stuttgart, Germany, 19–21 September 2023; pp. 1–8.

38. Stefani, T.; Christensen, J.M.; Hoemann, E.; Anilkumar Girija, A.; Köster, F.; Krüger, T.; Hallerbach, S. Applying Model-Based System Engineering and DevOps on the Implementation of an AI-based Collision Avoidance System. In Proceedings of the 34th Congress of the International Councilof the Aeronautical Sciences (ICAS), Florence, Italy, 9–13 September 2024; pp. 1–12.
39. Anilkumar Girija, A.; Christensen, J.M.; Stefani, T.; Hoemann, E.; Durak, U.; Köster, F.; Hallerbach, S.; Krüger, T. Towards the Monitoring of Operational Design Domains Using Temporal Scene Analysis in the Realm of Artificial Intelligence in Aviation. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 29 September–3 October 2024; pp. 1–8. https://doi.org/10.1109/dasc62030.2024.10749445.
40. Christensen, J.M.; Anilkumar Girija, A.; Stefani, T.; Durak, U.; Hoemann, E.; Köster, F.; Krüger, T.; Hallerbach, S. Advancing the AI-Based Realization of ACAS X Towards Real-World Application. In Proceedings of the 2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI), Herdon, VA, USA, 28–30 October 2024; pp. 57–64. https://doi.org/10.1109/ICTAI62512.2024.00017.
41. Christensen, J.; Hoemann, E.; Köster, F.; Hallerbach, S. Defining Operational Conditions for Safety-Critical AI-Based Systems from Data. *arXiv* **2026**, arXiv:2601.22118. https://doi.org/10.48550/ARXIV.2601.22118.
42. Torens, C.; Juenger, F.; Schirmer, S.; Schopferer, S.; Zhukov, D.; Dauer, J.C. Ensuring Safety of Machine Learning Components Using Operational Design Domain. In Proceedings of the AIAA SciTech 2023 Forum, National Harbor, MD, USA, 23–27 January 2023; p. 1124.
43. Gariel, M.; Shimanuki, B.; Timpe, R.; Wilson, E. Framework for Certification of AI-Based Systems. *arXiv* **2023**, arXiv:2302.11049.
44. Hasterok, C.; Stompe, J.; Pfrommer, J.; Usländer, T.; Ziehn, J.; Reiter, S.; Weber, M.; Riedel, T. *PAISE®. Das Vorgehensmodell für KI-Engineering*; Technical Report; Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB): Karlsruhe, Germany, 2021.
45. Zhang, R.; Albrecht, A.; Kausch, J.; Putzer, J, H.; Geipel, T.; Halady, P. DDE process: A requirements engineering approach for machine learning in automated driving. In *Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, 20–24 September 2021*; IEEE: New York, NY, USA, 2021; pp. 269–279.
46. Christensen, J.M.; Stefani, T.; Anilkumar Girija, A.; Hoemann, E.; Vogt, A.; Werbilo, V.; Durak, U.; Köster, F.; Krüger, T.; Hallerbach, S. Formulating an Engineering Framework for Future AI Certification in Aviation. *Aerospace* **2025**, *12*, 482. https://doi.org/10.3390/aerospace12060482.
47. Cappi, C.; Cohen, N.; Ducoffe, M.; Gabreau, C.; Gardes, L.; Gauffriau, A.; Ginestet, J.B.; Mamalet, F.; Mussot, V.; Pagetti, C.; et al. How to design a dataset compliant with an ML-based system ODD? In Proceedings of the 12th European Congress on Embedded Real Time Software and Systems (ERTS), Toulouse, France, 11–12 June 2024; pp. 1–10. https://doi.org/10.48550/ARXIV.2406.14027.
48. Höhndorf, L.; Dmitriev, K.; Vasudevan, J.K.; Subedi, S.; Klarmann, N.; Holzapfel, F. Artificial Intelligence Verification Based on Operational Design Domain (ODD) Characterizations Utilizing Subset Simulation. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 29 September–3 October 2024; pp. 1–10. https://doi.org/10.1109/DASC62030.2024.10749586.
49. *WG-114*; AI Standards in Aviation. ICAO: Paris, France, 2021.
50. International, S. G-34 Artificial Intelligence in Aviation. Available online: https://standardsworks.sae.org/standards-committees/g-34-artificial-intelligence-aviation (accessed on 10 January 2025).
51. G-34 Artifical Intelligence In Aviation Committee. Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI. Available online: https://www.sae.org/standards/content/arp6983/ (accessed on 16 November 2024).
52. *ANSI/AIAA G-043B-2018*; Guide to the Preparation of Operational Concept Documents. American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2025.
53. Julian, K.D.; Kochenderfer, M.J. Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019; pp. 1–10. https://doi.org/10.1109/dasc43569.2019.9081748.
54. Christensen, J.M.; Zaeske, W.; Beck, J.; Friedrich, S.; Stefani, T.; Girija, A.A.; Hoemann, E.; Durak, U.; Köster, F.; Krüger, T.; et al. Towards Certifiable AI in Aviation: A Framework for Neural Network Assurance Using Advanced Visualization and Safety Nets. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 29 September–3 October 2024; pp. 1–9. https://doi.org/10.1109/dasc62030.2024.10749321.
55. Damour, M.; De Grancey, F.; Gabreau, C.; Gauffriau, A.; Ginestet, J.B.; Hervieu, A.; Huraux, T.; Pagetti, C.; Ponsolle, L.; Clavière, A. Towards Certification of a Reduced Footprint ACAS-Xu System: A Hybrid ML-Based Solution. In *Proceedings of the Computer Safety, Reliability, and Security, York, UK, 8–10 September 2021*; Springer: Cham, Switzerland, 2021; pp. 34–48. https://doi.org/10.1007/978-3-030-83903-1_3.
56. *ISO/IEC/IEEE 29148:2018(E)*; ISO/IEC/IEEE International Standard—Systems and software engineering—Life Cycle Processes—Requirements Engineering. ISO: Geneva, Switzerland, 2018; pp. 1–104. https://doi.org/10.1109/IEEESTD.2018.8559686.

57. Verein zur Förderung der internationalen Standardisierung von Automatisierungs- und Meßsystemen (ASAM) e.V. ASAM OpenODD Base Standard 1.0.0 Specification, 2025. https://publications.pages.asam.net/standards/ASAM_OpenODD/ASAM_OpenODD/latest/specification/index.html.
58. Mercedes-Benz Research & Development North America, Inc. Introducing DRIVE PILOT: An Automated Driving System for the Highway. Available online: https://group.mercedes-benz.com/dokumente/innovation/sonstiges/2023-03-06-vssa-mercedes-benz-drive-pilot.pdf (accessed on 7 November 2024).
59. Shakeri, A. Formalization of Operational Domain and Operational Design Domain for Automated Vehicles. In Proceedings of the 2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C), Cambridge, UK, 1–5 July 2024; pp. 990–997. https://doi.org/10.1109/QRS-C63300.2024.00131.
60. Weiss, G.; Zeller, M.; Schoenhaar, H.; Fraunhofer, C.D.; Kreutz, A. Approach for Argumenting Safety on Basis of an Operational Design Domain. In Proceedings of the 2024 IEEE/ACM 3rd International Conference on AI Engineering—Software Engineering for AI (CAIN), Lisbon, Portugal, 14–15 April 2024; pp. 184–193.
61. Adedjouma, M.; Botella, B.; Ibanez-Guzman, J.; Mantissa, K.; Proum, C.M.; Smaoui, A. Defining Operational Design Domain for Autonomous Systems: A Domain-Agnostic and Risk-Based Approach. In Proceedings of the SOSE 2024—19th Annual System of Systems Engineering Conference, Tacoma, WA, USA, 23–26 June 2024; pp. 166–171. https://doi.org/10.1109/SOSE62659.2024.10620936.
62. Lee, C.W.; Nayeer, N.; Garcia, D.E.; Agrawal, A.; Liu, B. Identifying the Operational Design Domain for an Automated Driving System through Assessed Risk. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1317–1322. https://doi.org/10.1109/IV47402.2020.9304552.
63. Viola, N.; Corpino, S.; Fioriti, M.; Stesina, F. Functional analysis in systems engineering: Methodology and applications. In *Systems Engineering-Practice and Theory*; InTech: Rijeka, Croatia, 2012; pp. 71–96.
64. S-18 Aircraft and Sys Dev and Safety Assessment Committee. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*; SAE International: Machesney Park, IL, USA, 1996. https://doi.org/10.4271/ARP4761.
65. Herrmann, M.; Witt, C.; Lake, L.; Guneshka, S.; Heinzemann, C.; Bonarens, F.; Feifel, P.; Funke, S. Using ontologies for dataset engineering in automotive AI applications. In Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 14–23 March 2022; pp. 526–531. https://doi.org/10.23919/DATE54114.2022.9774675.
66. Systems, M.P. General Properties and Characteristics of Sensors. Available online: https://www.monolithicpower.com/en/learning/mpscholar/sensors/intro-to-sensors/general-properties-characteristics?srsltid=AfmBOooPonoNMaH1XZkUdaUvY50wL1EeRCkqxWEdjWBG8xttduX4OQ1V (accessed on 10 December 2024).
67. Dodge, S.; Karam, L. Understanding how image quality affects deep neural networks. In Proceedings of the 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, 6–8 June 2016; pp. 1–6. https://doi.org/10.1109/QoMEX.2016.7498955.
68. Jiang, L.; Wang, X. Dataset Constrution through Ontology-Based Data Requirements Analysis. *Appl. Sci.* **2024**, *14*, 2237. https://doi.org/10.3390/app14062237.
69. Paton, N.W.; Chen, J.; Wu, Z. Dataset Discovery and Exploration: A Survey. *Acm Comput. Surv.* **2023**, *56* 102. https://doi.org/10.1145/3626521.
70. da Cunha Davison, J.a.C.; Tostes, P.I.; Guerra Carneiro, C.A. Framework Architecture for AI/ML Data Management for Safety-Critical Applications. In Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 29 September–3 October 2024; pp. 1–9. https://doi.org/10.1109/DASC62030.2024.10748782.
71. Kreuzberger, D.; Kühl, N.; Hirschl, S. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE access* **2023**, *11*, 31866–31879.
72. Stanford Intelligent Systems Laboratory. *VerticalCAS Repository*; Stanford Intelligent Systems Laboratory: Stanford, CA, USA, 2020.
73. Stanford Intelligent Systems Laboratory. *HorizontalCAS Repository*; Stanford Intelligent Systems Laboratory: Stanford, CA, USA, 2020.
74. Christensen, J.M.; Anilkumar Girija, A.; Stefani, T.; Durak, U.; Hoemann, E.; Köster, F.; Krüger, T.; Hallerbach, S. *Advancing the AI-Based Realization of ACAS X Towards Real-World Application*, Version 1.0.0. Zenodo: Geneva, Switzerland, 2024. https://doi.org/10.5281/ZENODO.13815668.
75. RTCA, Inc. *Minimum Operational Performance Standards for Airborne Collision Avoidance System X (ACAS X) (ACAS Xa and ACAS Xo)*; Tech Report DO-385; GlobalSpec: Washington, DC, USA, 2018.
76. RTCA, Inc. *Minimum Operational Performance Standards for Airborne Collision Avoidance System Xu (ACAS Xu)*; Tech Report DO-386; GlobalSpec: Washington, DC, USA, 2020.
77. FlightGear Developers & Contributors. FlightGear. Available online: https://www.flightgear.org/ (accessed on 9 March 2023).
78. EUROCAE. *Minimum Aviation System Performance Standard for Detect and Avoid (Traffic) in Class A-C Airspaces—with Corrigendum 1*; Technical Report; EUROCAE: Saint-Denis, France, 2022.

79. EUROCAE. *ED-313*; Technical Report; EUROCAE: Saint-Denis, France, 2023.
80. Safety, S.A. GPWS. Available online: https://skybrary.aero/gpws (accessed on 28 December 2024).
81. Civil Aviation Authority of New Zealand. ADS-B in New Zealand. Available online: https://www.nss.govt.nz/assets/nss/resources/2018-07-31-ADSB-FAQ-Document-V0.3.docx.pdf (accessed on 10 January 2025).
82. Force, U.S.S. GPWS. Available online: https://www.spaceforce.mil/About-Us/Fact-Sheets/Article/2197765/global-positioning-system/ (accessed on 29 December 2024).
83. Oceanic, N.; Administration, A. GPS Accuracy. Available online: https://archive.gps.gov/systems/gps/performance/accuracy/ (accessed on 29 December 2024).
84. Chryssanthacopoulos, J.P.; Kochenderfer, M.J. Collision avoidance system optimization with probabilistic pilot response models. In Proceedings of the Proceedings of the 2011 American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 2765–2770. https://doi.org/10.1109/ACC.2011.5990776.
85. Julian, K.D.; Lopez, J.; Brush, J.S.; Owen, M.P.; Kochenderfer, M.J. Policy compression for aircraft collision avoidance systems. In Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 25–29 September 2016; pp. 1–10. https://doi.org/10.1109/DASC.2016.7778091.
86. Julian, K.D.; Kochenderfer, M.J.; Owen, M.P. Deep Neural Network Compression for Aircraft Collision Avoidance Systems. *J. Guid. Control. Dyn.* **2019**, *42*, 598–608. https://doi.org/10.2514/1.G003724.
87. Aniculaesei, A.; Schindler, C.; Knieke, C.; Rausch, A.; Rohne, D.; Richter, A. A Method for ODD Specification and Verification with Application for Industrial Automated Driving Systems. In *Proceedings of the 2023 International Conference on Computational Science and Computational Intelligence (CSCI)*; IEEE: New York, NY, USA, 2023; pp. 1519–1526.
88. de Grancey, F.; Gerchinovitz, S.; Alecu, L.; Bonnin, H.; Dalmau, J.; Delmas, K.; Mamalet, F. On the Feasibility of EASA Learning Assurance Objectives for Machine Learning Components. In Proceedings of the ERTS 2024, Toulouse, France, 11–12 June 2024; ERTS2024. Best paper award at ERTS 2024.
89. Wen, X.; Ma, H.L.; Chung, S.H.; Khan, W.A. Robust airline crew scheduling with flight flying time variability. *Transp. Res. Part E: Logist. Transp. Rev.* **2020**, *144*, 102132. https://doi.org/https://doi.org/10.1016/j.tre.2020.102132.
90. Eltoukhy, A.E.E.; Hashim, H.A.; Hussein, M.; Khan, W.A.; Zayed, T. Sustainable vehicle route planning under uncertainty for modular integrated construction: Multi-trip time-dependent VRP with time windows and data analytics. *Ann. Oper. Res.* **2025**, *348*, 863–898. https://doi.org/10.1007/s10479-024-06442-2.
91. Khan, W.A.; Ma, H.L.; Chung, S.H.; Wen, X. Hierarchical integrated machine learning model for predicting flight departure delays and duration in series. *Transp. Res. Part C Emerg. Technol.* **2021**, *129*, 103225. https://doi.org/https://doi.org/10.1016/j.trc.2021.103225.